Advant® OCS
with Master software

On-line Builder

Version 2.6

Reference Manual

3BSE 001 974R601

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

ABB Master, MasterAid, MasterBus, MasterPiece, and MasterView are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

MasterGate, MasterFieldbus, and MasterNet are trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

Advant, AdvaBuild, and AdvaInform are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

AdvaCommand, AdvaSoft, Advant Station, AdvaBuild On-line Builder, Advant Controller, and Advant Fieldbus are trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

IBM and all IBM-products mentioned in this publication are registered trademarks of International Business Machines Corp.

Microsoft, MS-DOS, and Windows are registered trademarks, Windows NT is a trademark of Microsoft Corporation.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrietechnik GmbH. ABB Industrietechnik GmbH assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrietechnik GmbH be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrietechnik GmbH be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrietechnik GmbH’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Industrietechnik GmbH 1999.

3BSE001264/E

Template: 3BSE001286/E

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General Information.......... 1-1
1.2 Manual Organization 1-1
1.3 Conventions 1-1
1.4 Related Documentation.......... 1-2
1.5 Terminology.......... 1-4
1.6 Differences between Environments.......... 1-4
1.7 Installation 1-6

Chapter 2 - Command Sets

2.1 How to Work with Command Sets 2-1
2.2 Command Sets together with DSPU131.......... 2-1
2.3 Command Sets together with RTA Board.......... 2-4
2.4 Command Sets together with Engineering Board PU516 2-5

Chapter 3 - Available Commands

3.1 AALU - Assign Alternate Logical Unit.......... 3-1
3.2 ACT - ACtivate Task 3-1
3.3 ADTIME - ADjust TIME 3-1
3.4 AINS - Add INStance.......... 3-2
3.5 ALU - Assign Logical Unit 3-3
3.6 ALUS - Assign Logical Unit to Sequential segment.......... 3-3
3.7 AMEMB - Add MEMBer.......... 3-3
3.8 ANPER - ANalyse PERformance.......... 3-4
3.9 ANSYS - ANalyse SYStem.......... 3-4
3.10 APP - Alter Product Parameters 3-4
3.11 BL - BLock.......... 3-5
3.12 BLRS - BLock and ReSet.......... 3-6
3.13 C - Connect.......... 3-6
3.14 CALCCHS - CALCulate CHeckSum.......... 3-7
3.15 CIP - Connect InPort.......... 3-8
3.16 CLLFILE - CLose LogFILE.......... 3-8
3.17 CLOSE - CLOSE segment 3-8
3.18 COMPARE - COMPARE segments 3-9
3.19 CONTINUE.......... 3-9
3.20 COP - Connect OutPort 3-9
3.21 COPY - COPY segment 3-10
3.22 COSESS - Connect Offline SESSion 3-10

CONTENTS (continued)

3.23 CRDB - CReate Data Base 3-10
3.24 CREATE - CREATE segment 3-11
3.25 CSPT - CompreSs Program Tables 3-11
3.26 CSSD - CompresS System Data 3-11
3.27 CU - Connect Undefined 3-12
3.28 DAABF 3-12
3.29 DBL - DeBLock 3-13
3.30 DCOP - DisConnect OutPort 3-13
3.31 DCTARG - DisConnect TARGet 3-13
3.32 DD - Delete Debug point 3-14
3.33 DDB - Delete Data Base 3-14
3.34 DDC - Delete Debug point Counter 3-15
3.35 DDCD - Delete Debug point Catch Data 3-15
3.36 DDCON - Delete Debug point CONdition 3-15
3.37 DDTD - Delete Debug point Trace Data 3-16
3.38 DDTSL - Delete Debug point control of Task Switch Log 3-16
3.39 DELAY 3-16
3.40 DELDOC 3-16
3.41 DELETE - DELETE segment 3-17
3.42 DELEVENT - DELete EVENTs and alarms 3-17
3.43 DFAD - DeFine Absolute Debug point 3-18
3.44 DFDC - DeFine Debug point Counter 3-18
3.45 DFDCD - DeFine Debug point Catch Data 3-19
3.46 DFDCON - DeFine Debug point CONdition 3-19
3.47 DFDT - DeFine Debug point Type 3-20
3.48 DFDTD - DeFine Debug point Trace Data 3-20
3.49 DFDTSL - DeFine Debug point control of Task Switch Log 3-21
3.50 DFLSYMB - DeFine Local SYMBol 3-21
3.51 DFNETUSE - DeFine NETwork USE 3-22
3.52 DFRD - DeFine Relative Debug point 3-22
3.53 DFSYMB - DeFine (global) SYMBol 3-23
3.54 DIBM - DIsable Build Mode 3-23
3.55 DICONFIG - DIsable CONFIGuration 3-24
3.56 DID - DIsable Debug point 3-24
3.57 DIIP - DIsable InPort 3-24
3.58 DIMDB - DIMension Data Base 3-25
3.59 DIMP - DIsable Modify Permission 3-26

CONTENTS (continued)

3.60 DIMPC - DIMension PC 3-26
3.61 DINS - Delete INStance 3-27
3.62 DITO - DIsable Time Out 3-27
3.63 DMEMB - Delete MEMBer 3-27
3.64 DS - Delete Statement 3-28
3.65 DSYMB - Delete (global) SYMBol 3-28
3.66 DUAP - DUmp APplication 3-29
3.67 DUCON - DUmp CONcept(s) 3-30
3.68 DUDB - DUmp Data Base 3-31
3.69 DUDBS - DUmp Data Base Source 3-31
3.70 DUMPFILE 3-32
3.71 DUPCS - DUmp PC Source 3-33
3.72 DURWM - DUmp RWM 3-33
3.73 DUTDB - DUmp Total Data Base 3-34
3.74 DUTPT - DUmp Total Program Tables 3-34
3.75 DV - Delete Values 3-34
3.76 EBM - Enable Build Mode 3-35
3.77 ECHO 3- 35
3.78 ECONFIG - Enable CONFIGuration 3-36
3.79 ED - Enable Debug point 3-36
3.80 EIP - Enable InPort 3-36
3.81 EMP - Enable Modify Permission 3-37
3.82 ETO - Enable Time Out 3-37
3.83 ETTY - Enable TTY mode 3-37
3.84 EXECFILE - EXEcute Command FILE 3-38
3.85 EXECSUB - EXEcute SUBroutine 3-38
3.86 EXIT 3-38
3.87 FDISP - Find DISPlay(s) 3-38
3.88 FINS - Find INStance 3-39
3.89 FNEX - Find NEXt Instance 3-39
3.90 GEPCD - GEnerate PC Diagram 3-39
3.91 GEPD - GEnerate Page Division 3-41
3.92 GETAB - GEnerate TABle of values 3-41
3.93 GFOR - Get FORmat 3-48
3.94 GID - Get IDentity 3-49
3.95 GINS - Get INStance 3-49
3.96 GLID - Get Last Item Designation 3-49

CONTENTS (continued)

3.97 GOTO 3-50
3.98 GPN - Get Page Number 3-50
3.99 GPRO - Get PROperty 3-50
3.100 HARDCOPY 3-50
3.101 HELP 3-51
3.102 IF 3-52
3.103 INCPT - INCrease Program Tables 3-52
3.104 INQUIRE 3-52
3.105 INSTASK - INStall TASK 3-53
3.106 INTERACTION 3-53
3.107 IS - Insert Statement 3-53
3.108 KERMIT 3-54
3.109 KIT - KIll Task 3-55
3.110 LAABF - Load AAb File 3-55
3.111 LAVOL - List AVailable VOLumes 3-55
3.112 LC - List Channels 3-55
3.113 LCI - List Channel Information 3-56
3.114 LCT - List Concept Table 3-56
3.115 LCU - List Channel Utilization 3-56
3.116 LD - List Debug points 3-57
3.117 LDBD - List Data Base Diagram 3-57
3.118 LDEV - List DEVices 3-58
3.119 LDEVI - List DEVice Information 3-58
3.120 LDIR - List DIRectory 3-59
3.121 LDISP - List DISPlay information 3-59
3.122 LDURWM - List info about RWM dump 3-60
3.123 LGAM - List Group Alarm Members 3-60
3.124 LIF - List Inhibit Flags 3-61
3.125 LIQB - List I/O Queue Blocks 3-61
3.126 LIST - LIST segment 3-61
3.127 LLUA - List Logical Unit Assignments 3-62
3.128 LMFS - List MasterFieldbus Status 3-62
3.129 LMP - List Modify Permission 3-62
3.130 LOADFILE 3-62
3.131 LOAP - LOad APplication 3-63
3.132 LOCK 3-64
3.133 LOCON - LOad CONcept(s) 3-64

CONTENTS (continued)

3.134 LOCPSET - LOad CP SET 3-65
3.135 LODB - LOad Data Base 3-65
3.136 LORWM - LOad RWM 3-66
3.137 LOTDB - LOad Total Data Base 3-66
3.138 LOTPT - LOad Total Program Table 3-66
3.139 LPCD - List PC Diagram 3-67
3.140 LPCL - List PC List 3-67
3.141 LPI - List Port Information 3-68
3.142 LPN - List Page Number 3-68
3.143 LS - List Statement 3-68
3.144 LSIG - List SIGnals 3-69
3.145 LSYSHI - List SYStem Halt Information 3-70
3.146 LSS - List Special Status 3-70
3.147 LSYSM - List SYStem Messages 3-71
3.148 LT - List Tasks 3-71
3.149 LTI - List Task Information 3-71
3.150 LTIMQ - List TIMer Queues 3-72
3.151 LTREE - List TREE 3-72
3.152 LTRQ - List Task Ready Queues 3-73
3.153 LTSL - List Task Switch Log 3-73
3.154 LV - List Variable 3-73
3.155 MDB - Modify Data Base 3-74
3.156 MDT - Modify Document Texts 3-74
3.157 MFOR - Modify FORmat 3-75
3.158 MINS - Modify INStance 3-75
3.159 MNV - Modify Name of Variable 3-75
3.160 MPD - Modify Page Division 3-76
3.161 MPH - Modify Page Headline 3-77
3.162 MPN - Modify Page Number 3-78
3.163 MPRO - Modify PROperty 3-78
3.164 MSTABS - Modify Symbol TABle Size 3-79
3.165 MV - Modify Value 3-79
3.166 MVD - MoVe Debug point 3-79
3.167 NEWPAGE 3-80
3.168 NEWDOC 3-80
3.169 NOON 3- 80
3.170 O - Open 3-80

CONTENTS (continued)

3.171 OBIT - Open BIT 3-81
3.172 OLFILE - Open LogFILE 3-82
3.173 ON 3-82
3.174 OPEN - OPEN segment 3-83
3.175 OPS - Open Program Status register 3-83
3.176 OREG - Open REGister 3-83
3.177 OTC - Open Task Code 3-84
3.178 OTDB - Open Task Data relative Base level 3-85
3.179 OTDC - Open Task Data relative Current level 3-85
3.180 PD - Proceed from Debug point 3-86
3.181 PGPLC - ProGram PLC unit 3-87
3.182 POPDB - Data Base POPulation 3-87
3.183 PROM - set up a PROM session 3-88
3.184 R - Read 3-89
3.185 RDATIME - Read DAte and TIME 3-89
3.186 RECONFIG - REstart CONFIGuration 3-90
3.187 RECV - RECeiVe 3-90
3.188 RENAME - RENAME segment 3-91
3.189 RESTART - load prommed application and RESTART system 3-91
3.190 RET - REsume Task 3-91
3.191 RETURN 3-91
3.192 RLFILE - Read LogFILE 3-92
3.193 RLMD - Read Local Motor Drive 3-92
3.194 RMD - Remove Display 3-92
3.195 RREG - Read REGisters 3-93
3.196 RSIF - ReSet Inhibit Flag 3-93
3.197 RTC - Read Task Code 3-93
3.198 RTDB - Read Task Data relative Base level 3-94
3.199 RTDC - Read Task Data relative Current level 3-94
3.200 RTIME - Read TIME 3-95
3.201 RUDAS - Retrieve User Disk Application Segments 3-95
3.202 SDATIME - Set DAte and TIME 3-95
3.203 SEIF - SEt Inhibit Flag 3-96
3.204 SELNM - SELect Network Management 3-96
3.205 SEND 3-96
3.206 SEOM - SEt Output Medium 3-97
3.207 SEPS - SEt Page Size 3-97

CONTENTS (continued)

3.208 SERN - SEt Real Notation 3-97
3.209 SHLSYMB - SHow Local SYMBol 3-98
3.210 SHSYMB - SHow (global) SYMBol 3-98
3.211 SHTARG - SHow TARGet 3-98
3.212 SHUDAS - SHow User Disk Application Segments 3-98
3.213 SLG - SeLect Graphical mode 3-99
3.214 SLLEV - SeLect command LEVel 3-99
3.215 SLNG - SeLect Non-Graphical mode 3-100
3.216 SLNUM - SeLect NUMeric identities 3-100
3.217 SLSYM - SeLect SYMbolic identities 3-100
3.218 SLTARG - SeLect TARGet 3-100
3.219 STIME - Set TIME 3-101
3.220 SUDAS - Save User Disk Application Segments 3-101
3.221 SUSPEND 3-101
3.222 SWBL - SWitch BLockings 3-102
3.223 TRDBS - TRanslate Data Base Source 3-102
3.224 TRPCS - TRanslate PC Source 3-103
3.225 TSESS - Terminate SESSion 3-104
3.226 TYPE 3-104
3.227 UHR - Update Harddisk Remotedirectory files 3-104
3.228 UNCSSD - UNCompresS System Data 3-105
3.229 UNLOCK 3-105
3.230 USERINPUT 3-105
3.231 VERIFY 3-105
3.232 W - Write 3-106
3.233 WLFILE 3-106
3.234 WTC - Write Task Code 3-106
3.235 WTDB - Write Task Data relative Base level 3-107
3.236 WTDC - Write Task Data relative Current level 3-108

CONTENTS (continued)

Chapter 4 - Special Commands for HP-UX Environment

4.1 dobackup 4-1
4.2 dorestore 4-1
4.3 PUbackup 4-1
4.4 PUboards 4-2
4.5 PUlock 4-2
4.6 PUrelease 4-2
4.7 PUrestore 4-2
4.8 PUvoldef 4-2
4.9 rta_boot 4-2

Chapter 5 - Working Methods of the On-line Builder

5.1 Volumes and Segments 5-1
5.2 Wildcards 5-2
5.3 Default Volume and Segment Names 5-2
5.4 Levels in On-line Builder 5-3
5.5 Targets in a Network Environment 5-3

Chapter 6 - Command Files

6.1 Building a Command File 6-1
  6.1.1 User Commands 6-1
  6.1.2 Comments 6-2
  6.1.3 Commands to Control the Execution 6-2
    6.1.3.1 Commands to Control Input and Output 6-3
    6.1.3.2 Symbols in Command Files 6-6
    6.1.3.3 Commands to Control the Execution Flow 6-6
  6.1.4 Example 6-9
6.2 Executing a Command File 6-10
  6.2.1 EXECFILE - EXEcute Command FILE 6-10
  6.2.2 EXECSUB - EXEcute SUBroutine 6-12
6.3 Handling Errors 6-13
  6.3.1 Error Codes 6-13
  6.3.2 Special Procedures 6-13
6.4 Predefined Symbols for Command Menus 6-14
  6.4.1 Graphic Presentation of Menus 6-14
  6.4.2 Return to specified Cursor Position 6-18
  6.4.3 Error Displays on Answer Lines 6-22

CONTENTS (continued)

6.5 User-defined Symbols.. 6-23
- 6.5.1 General.. 6-23
- 6.5.2 Global and Local Symbols.. 6-23
- 6.5.3 Defining Symbols.. 6-23
- 6.5.4 Displaying Symbol Values.. 6-25
- 6.5.5 Deleting Symbols.. 6-25
- 6.5.6 Different Expression Types to define Symbols 6-26
  - 6.5.6.1 Character Strings.. 6-26
  - 6.5.6.2 Integers and Real Numbers.. 6-27
  - 6.5.6.3 Lexical Functions.. 6-27
  - 6.5.6.4 How to use Lexical Functions.. 6-28
  - 6.5.6.5 Expressions with Operators 6-30

ILLUSTRATIONS

Figure 3-1. Display of GETAB in graphical mode 3-43
Figure 3-2. Value List window.. 3-44
Figure 3-3. Setup window.. 3-44
Figure 3-4. Time Scale dialog.. 3-46
Figure 3-5. Trigger Setup dialog.. 3-47
Figure 6-1. Graphic Presentation of Command File Menus 6-15
Figure 6-2. Command File Main Menu with Text 6-16
Figure 6-3. Command File Main Menu with Submenus.. 6-17
Figure 6-4. Screen Display when giving an Illegal Answer 6-19
Figure 6-5. Screen Display when giving a Valid Answer 6-21
Figure 6-6. Screen Display when giving an Illegal Answer and using MA$ILLEGAL6-22

TABLES

Table 1-1. Keyboard Interaction.. 1-1
Table 1-2. Related Documentation 1-2
Table 1-3. Terms associated with On-line Builder 1-4
Table 1-4. Function keys 1-5
Table 2-1. Available Command Sets 2-1
Table 6-1. Description of Comparison Operators.. 6-7
Table 6-2. Currently available Error Codes.. 6-13
Table 6-3. Overview of Operators in Expressions.. 6-30

Chapter 1 Introduction

1.1 General Information

This manual describes On-line Builder functions for the software products AMPLControl Configuration and AdvaCommand.

It gives a summary of commands available and an introduction on basic On-line Builder features as file handling commands and building command files.

1.2 Manual Organization

This chapter, Chapter 1, Introduction, describes the conventions used in this manual and general characteristics of the On-line Builder. Chapter 2.1, Command Sets explains the start up procedures of the On-line Builder.

Chapter 3, Available Commands and Chapter 4, Special Commands for HP-UX Environment give systematically explanations on all available commands in alphabetical order.

Chapter 5, Working Methods of the On-line Builder and Chapter 6, Command Files explain how to work with the On-line Builder and how to build and use command files.

1.3 Conventions

Keyboard Interaction

Table 1-1. Keyboard Interaction

Read	Press
<F4>	key labeled **F4**
<Ctrl-Break>	**Ctrl** key while pressing the key labeled **Break**

Syntax

The syntax of each command is described in the form of the command name, followed by the arguments that are to be specified. The arguments are separated with symbols such as comma “,”, slash “/”, semicolon “;” or right angle bracket “>”. To carry out a command, enter the command syntax and press the **Enter** key.

- Bold strings are the command names and arguments; they must be typed as they appear.
- Italic strings represent substitutable arguments.
- Brackets “[ ]” around an argument name indicate that the argument is optional.
- Arguments, to be used alternatively, are indicated by a vertical bar (in some syntax descriptions alternative choices are represented by an italic string and a separate explanation of its substitutions).

Examples:

```
LTREE item_des[;number_of_elements][/mode]
LDBD SEL=select[=start][,interv][,gap][/XREF[;PCm[>PCn]]]
```

The following conventions are used in examples of screen dialogs:

- Bold strings represents input from the user.
- Normal strings represents dialog text from the target system or the engineering station.

1.4 Related Documentation

Below is a list of related documents:

Table 1-2. Related Documentation

Manual	Description
Advant Station 100 Series Engineering Station Environment User’s Guide	Describes the Advant Station 100 Series Engineering Stations’ hardware.
Application Builder User’s Guide	Describes Application Builder, Bus Configuration Builder and On-line Builder start-up.
On-line Builder User’s Guide	Describes the Windows functions of On-line Builder. Besides the detailed description of all runtime operation menus and dialogs you find also important information about special print functions, start-up procedures and command line editing functions.

Table 1-2. Related Documentation (Continued)

Manual	**Description**
AMPL Configuration Advant Controller 400 Series Reference Manual	Contains instructions for configuration and application programming of AC 410/450 controllers including command descriptions. The topics are: AMPL configuration principals, application programming, commissioning and testing, on-line editing, documentation, backup and transfer of application data
Advant Controller 410 User's Guide	Contains a description of Advant Controller 410 and its functions.
Advant Controller 450 User's Guide	Contains a description of Advant Controller 450 and its functions.
Advant Station 500 Series Operator Station User's Guide	Manual for Advant Station 500 Series OS.
Advant Controller 400 Series PC Elements Reference Manual	Data for all PC elements in Advant Controller 410/450.
Advant Controller 400 Series Data Base Elements Reference Manual	Data for all data base elements in Advant Controller 410/450.
ABB Master Source Code Handling	Contains descriptions and instructions for use of AMPL source code for programs and data bases.
MasterFieldbus and S400 I/O User's Guide	Complete manual on MasterFieldbus and the S400 I/O system. Contains technical descriptions, instructions for installation, commissioning fault tracing and technical data. This manual also describes MasterPiece 51 as a distributed unit, i.e. the engineering required in AC 410/450, MasterPiece 200/1 and the programming.

1.5 Terminology

The following is a list of terms associated with On-line Builder.

Table 1-3. Terms associated with On-line Builder

Term	Description
Advant Station 100 Series Engineering Stations	Common term for Advant Station 120 Engineering Station AS 120 ES), and Advant Station 130 Engineering Station (AS 130 ES).
Advant Station 500 Series Engineering Stations	Common term for Advant Station 515 Engineering Station and Advant Station 520i Engineering Station.
Advant Controller 400 Series	Common term for Advant Controller 410 and Advant Controller 450.
DSPU131	Advant Station 100 Series Engineering Board; ISA Board
RTA Board	Real Time Accelerator Board used with AdvaCommand on HP-UX and Windows NT
PU516	Advant Station 100 Series Engineering Board; PCI Board
Engineering Board	RTA Board, DSPU131 or PU516

1.6 Differences between Environments

On-line Builder can run in different environments. Environments can be destinguished by the operating system or by the engineering board.

On-line Builder can be based on Windows NT and HP-UX.

With Windows NT, the On-line Builder can use the Engineering Board type DSPU131 or PU516. If used together with AdvaCommand for Windows NT, the RTA board is used.

With HP-UX, it can use the RTA Board. Depending on the used board On-line Builder has certain differences, which are described below.

Function Keys

Several function keys are available in Windows NT , but not in HP-UX environment:

Table 1-4. Function keys

Windows NT use the key	**HP-UX use the key**	**To**
<**Ins**>	<**Ctrl+A**>	Toggle between insert and overwrite mode. The default editing mode is the insert mode.
<**Del**>	<**Delete character**>	Delete the character at the cursor’s position
<**Home**>	no key available	Move the cursor to the first position in the command line
<**End**>	no key available	Move the cursor to the last position in the command line
<**Left arrow**>	<**Left arrow**>	Move the cursor left
<**Right arrow**>	<**Right arrow**>	Move the cursor right
<**Ctrl+left arrow**>	<**Ctrl+left arrow**>	Move to the next word left
<**Ctrl+right arrow**>	<**Ctrl+right arrow**>	Move to the previous word right
<**Back space**>	<**Back space**>	Delete the character left of the cursor
<**Alt**>	no key available	Stop scrolling of listings
<**Alt+up arrow**>	<**Ctrl+up arrow>**	Scroll one line upwards
<**Alt+down arrow**>	<**Ctrl+down arrow**>	Scroll one line downwards
<**Alt+page up**>	<**Ctrl+page up**>	Scroll one page upwards
<**Alt+page down**>	<**Ctrl+page down**>	Scroll one page downwards

Keyboard

Depending on the keyboard, some keys may be named different.

Examples:

- <**Backspace**> or just an arrow
- <**Enter**> or <**Return**>

Use in these cases the corresponding key instead.

Commands

The following On-line Builder commands are not available with the DSPU131 board (see Chapter 3, Available Commands):

```
COSESS, DFNETUSE, LOCPSET
```

The following On-line Builder commands are not available with the Engineering Board type PU516 (see Chapter 3, Available Commands):

```
ALUS, CALCCHS, CLOSE, CSSD, DURWM, ETTY, HARDCOPY, INSTASK,
KERMIT, LDURWM, LORWM, LSYSHI, LSYSM, OPEN, PROM, RECV, SEND,
SLNM (expert level), UHR, UNCSSD
```

The following AdvaBuild On-line Builder commands are not available with the RTA board (see also Chapter 4, Special Commands for HP-UX Environment):

```
ALUS, ANSYS, CALCCHS, CLOSE, CSSD, DURWM, ETTY, HARDCOPY,
INSTASK, KERMIT, LCU, LDURWM, LIQB, LORWM, LSIG, LSYSHI, LSYSM,
LTIMQ, LTRQ, LTSL, OPEN, PROM, RECV, SEND, SLNM (expert level),
UHR, UNCSSD
```

Quitting Commands

The subcommand <**Ctrl-Break**> is not implemented in HP-UX environment. Use the key combination <**Ctrl-B**> instead.
In Windows NT environment only: The key combination <**Alt-B**> can be used alternatively to <**Ctrl-Break**>.

Selecting Command Sets

Depending on the type of the engineering board selecting of command sets works different. For more information see Chapter 2, Command Sets.

1.7 Installation

For installation purposes refer to the platform specific user’s guides and Release Notes.

Chapter 2 Command Sets

2.1 How to Work with Command Sets

The On-line Builder commands are grouped to a series of command sets. A set includes all commands necessary to work with a dedicated target like Advant Controller 400 or Advant Station 520 Operator Station. To start a session, select one of the sets from a menu.

During the session the commands of the selected command set are available.

To load a new set of commands, terminate the session wit the command TSESS, load the required command set and start the session again. When selecting another set, the previous command set will be exchanged by the new one.

Normally, On-line Builder runs in Automatic Startup mode. In this mode On-line Builder selects the commands sets that fit to the selected node. If you want to select the commands youself, run the **Options | Setup** menu command and select the corresponding option. The available command sets are described in Section 2.2, Command Sets together with DSPU131 for On-line Builder with Engineering Board type DSPU131 and following sections for the RTA board and the Engineering Board PU516.

2.2 Command Sets together with DSPU131

Table 2-1. Available Command Sets

Menu Supported ABB Master Products	**Submenu Available Activities**
1. Advant Controller 400	1. Advant Controller 400 Programming 2. Advant Controller 400 Programming via Network 3. Advant Controller 400 Blocking on Overload 4. Post Mortem Analysis 5. Switching Running PC Programs
2. Advant Station 500 Operator Station	
3. Advant Station 500 Engineering Station	
4. MasterPiece 200	1. MasterPiece 200 Programming 2. MasterPiece 200 Programming via Network 3. MasterPiece 200 Blocking on Overload

Table 2-1. Available Command Sets (Continued)

Menu Supported ABB Master Products	Submenu Available Activities
	4. Post Mortem Analysis
	5. Switching Running PC Programs
5. MasterView 800	1. MasterView 800 Programming 2. MasterView 800 Programming via Network 3. Post Mortem Analysis
6. MasterBus 300	1. Network Layman Commands 2. Network Expert Commands
7. MasterGate 230	1. MasterGate 230 Programming 2. MasterGate 230 Programming via Network 3. Post Mortem Analysis
8. MasterPiece 51	1. MasterPiece 51 Programming locally 2. MasterPiece 51 Programming via MasterFieldbus
9. Advant Station 100 Engineering Station	
10. More ...	1. Expert Commands 2. AccuRay board handling 3. User defined Set 1 (DEAO01A0) 4. User defined Set 2 (DEAO02A0) 5. User defined Set 3 (DEAO03A0) 6. User defined Set 4 (DEAO04A0) 7. User defined Set 5 (DEAO05A0)

The command selection menu consists of a two level menu. On the first level labelled **Menu** the targets you can work with are listed. On the second level, **Submenu**, you can specify a certain way of working with the selected target, for example, via Network.

Menu is firstly displayed. You are asked to select one target type by entering the number written to the left of the target type. Afterwards the corresponding **Submenu** will be displayed and you can select the working mode. Table 2-1 displays both menus together. For some targets no submenu is available.

Connection via Network

Select connection to a target via Network if you want to work with another target than your engineering station is physically connected to. You are first asked to enter the network and node number. Afterwards the Advant Station will be remotely connected to the corresponding target.

Blocking on Overload

Select this working mode if the target has come to a halt due to an overload situation. It makes it possible to block all PC programs with the BL command.

Post Mortem Analysis

With this working mode you can analyze a halted target.

Switching Running PC Programs

With this commands set you can switch the execution from one program to another one. Afterwards the running program will be blocked and the blocked one will take over.

Expert Commands

With this command set you can select special maintenance commands. They are for ABB Fieldservice personnel only. These commands are not part of the standard products, but of the software package “Expert Utilities”.

User defined Selection

With this working mode you can select a user defined command set created with theMS-DOS command **ADDCP** (see Technical Information Binders). This mode is for ABB Fieldservice personnel only.

2.3 Command Sets together with RTA Board

When running together with the RTA Board, the On-line Builder presents an additional menu, called **Command Sets**. Select one of the items to start loading the corresponding set of commands.

For a description of the command sets see Section 2.2, Command Sets together with DSPU131. The following sets are not supported when using the RTA Board:

- Advant Controller 400
  - Advant Controller 400 Blocking on Overload
  - Post Mortem Ananysis
- MasterPiece 200
  - MasterPiece 200 Blocking on Overload
  - Post Mortem Ananysis
- MasterView 800
  - Post Mortem Analysis
- MasterBus 300
  - Network Expert Commands
- MasterGate 230
  - Post Mortem Analysis

Connection via Network

When working with the Real Time Accelerator Board a connection to a target is always done via Network, because the connection is performed viaMasterBus 300.

2.4 Command Sets together with Engineering Board PU516

When running together with the Engineering Board type PU516, the On-line Builder presents an additional menu, called **Command Sets**. Select one of the items to start loading the corresponding set of commands.

For a description of the command sets see Section 2.2, Command Sets together with DSPU131. The following sets are not supported when using the Engineering Board type PU516:

- MasterBus 300
    - Network Expert Commands

Connection via Network

When working with the Engineering Board type PU516, a connection to a target system can be done via MasterBus 300 network or via serial connection. The PU516 will first try to connect via MasterBus 300. If this fails, it tries to reach the target system via serial connection.

Command sets which require that the target system is halted, cannot be used when connecting to the target system via MasterBus 300.

Connection to halted target systems

To connect to a halted target system with the Engineering Board type PU516, use the command “COSESS”. This command establishes a session with the halted system using the serial connection.

User defined Selection

With this working mode you can select a user defined command set created with the a text editor. This mode is for ABB Fieldservice personnel only.

Chapter 3 Available Commands

The following descriptions give an overview of all available commands.

For each command it is noticed on which level it is available (PC level, system level). If a command is not part of the standard On-line Builder, but of Expert Utilities, it is also mentioned.

Some commands need deep knowledge about the Master system. It may be dangerous to use them. These commands are marked with the following warning:

WARNING

This command shall be used by experts only!

3.1 AALU - Assign Alternate Logical Unit

Syntax: **AALU** *logical_unit=alternate_logical_unit*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command assigns an alternate logical unit for a logical unit. “logical_unit” is original assignment (an integer). “alternate_logical_unit” is its alternative (an integer).

The system will try the alternate unit if the ordinary unit fails, provided the requesting task has specified system recovery in the request.

Examples: To assign logical unit 7 as alternative for logical unit 1:

```
AALU 1=7
```

3.2 ACT - ACtivate Task

Syntax: **ACT** *task_name*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command activates a task. Type HELP TASK_NAME for description of task names.

Examples: To activate the task DEFS003:

```
ACT DEFS003
```

3.3 ADTIME - ADjust TIME

Syntax: **ADTIME** *offset*

This command is available at PC and system level.

Description:

This command adjusts system time by adding a specified offset. “offset” is offset in seconds (-100..100). The correction is effected over a period of time, rather than immediately, to ensure continuity in system time. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: To set system time back by 70 seconds:

```
ADTIME -70
```

Example 2: To advance system time by 35 seconds:

```
ADTIME 35
```

3.4 AINS - Add INStance

Syntax: **AINS** *logic_file, logic_rec*

```
AINS logic_file,logic_rec;key or var
```

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command adds an instance to the data base. If the concept has a variant part on the base level, the variant to be added must be defined. Type HELP KEY for the description of key condition.

Example 1: Add instance which does not contain a variant part

```
AINS 11,1
```

Example 2: Add instance which contains a variant part

```
AINS 11,1;9=2
```

3.5 ALU - Assign Logical Unit

Syntax: **ALU** *logical_unit=device*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command assigns a logical unit to a device. “logical_unit” is logical unit (an integer). “device” is device assigned (1 to 6 characters).

Example: To assign logical unit 2 to device CONSOL:

```
ALU 2=CONSOL
```

3.6 ALUS - Assign Logical Unit to Sequential segment

Syntax: **ALUS** *logical_unit*=[*volume*]:*segment*,*access*

WARNING

This command shall be used by experts only!

This command is available at PC and system level.

Description: This command assigns a logical unit number to a segment. “logical_unit” is logical unit number. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name. “access” is access privilege (R for Read; W for Write).

This command is not available with the Engineering Board PU516 and the RTA board.

Example: To assign logical unit 6 to segment A.SA on volume XXXX:

```
ALUS 6=XXXX:A.SA,W
```

3.7 AMEMB - Add MEMBer

Syntax: **AMEMB** *mlevref*

```
AMEMB mlevref;key or variant
```

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command adds a member to a group. The last property of the multi level reference must be of group type or format reference. If no member number is defined with the property, the new member is added last in the group. Type HELP MLEVREF for the description of mlevref parameter. Type HELP KEY for the description of key condition.

3.8 ANPER - ANalyse PERformance

Syntax: **ANPER**

This command is available at PC and system level.

Description: This command measures the system load, the task load or the channel utilization of an Advant Controller 400 Series or an Advant Station 500 Series Operator Station. The options within this command can be selected by use of interactive menu selections. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.9 ANSYS - ANalyse SYStem

Syntax: **ANSYS**

This command is available at PC and system level.

Description: This command performs a thoroughly analyse of a halted system. A lot of functions are performed - other commands are called, some of them may belong to other command processors. As operator - be prepared on that a quite large list will be produced.

This command is not available with the RTA board.

3.10 APP - Alter Product Parameters

Syntax: **APP**

This command is available at PC and system level.

Description: You can use the **APP** command to change the cycle times of the interpreters.The command **APP** (Alter Product Parameter) presents and/or alters miscellaneous cycle (scan) times in the connected Advant Controller 400 Series.

Use the **APP** command to change to a suitable cycle time if the default setting is not satisfactory. Execute the **APP** command prior to all application programming.

Subcommands:

M	The line at the cursor (>) is opened for modification.
>	The cursor moves down one step after pressing **Enter**.
<	The cursor moves up one step after pressing **Enter**.
=	The cursor does not move after pressing **Enter**.
!	The subcommand “**!**” is used to switch from the modify mode to the initial mode.
DIM	The requested dimensioning is transferred to the Advant Controller 400 Series with the subcommand **DIM**.
PRINT	Gives a printout of the screen contents.
QUIT	To leave the **APP** command without any alterations.
<Ctrl-Break>	Same function as **QUIT**.
END	To leave the **APP** command while altering the cycle times.

If the cycle times in the connected Advant Controller 400 Series can be altered, the screen display below is provided. Otherwise, a similar presentation; only screen display but no dialogs are provided.

Allowed values for interpreter A:
5 ms ≤ scan_A ≤ 50 ms

Allowed values for interpreter B:
4*scan_A (25 ms max.) ≤ scan_B ≤ scan_C/4 (995 ms max.)

Allowed values for interpreter C:
4*scan_B (105 ms max.) ≤ scan_C ≤ 4000 ms

Allowed values for interpreter D:
1 s ≤ scan_D ≤ 6 s

Allowed values for DSC:
50 ms ≤ DSC ≤ 1000 s

Allowed values for MVI_DSC:
500 ms ≤ MVI_DSC ≤ 1000 s

Allowed values for DSP:
32, 64, 128, 256, 512, 1024, 2048, 4096ms

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.11 BL - BLock

Syntax: **BL**

```
BL [DB,] pc_pgm [> pc_pgm]
BL [DB,] item_des
```

This command is available at PC level only.

Description: This command can be used to:

- Block one execution unit (PCPGM, MASTER, CONTRM, SEQ) or all execution units in one or more PC programs.
- Block output to the data base (but not execution) from the specified execution unit or PC program(s).

If a PC program is already blocked, a command like **BL DB, PCnn** will start the PC program, but with output to the data base blocked.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Block all execution of execution unit PC1.1:

```
#BL PC1.1
```

Example 2: Block output to the data base from execution unit PC1.2:

```
#BL DB,PC1.2
```

Example 3: Block all execution of PC2, PC3 and PC4:

```
#BL PC2>PC4
```

3.12 BLRS - BLock and ReSet

Syntax: **BLRS** [**DB**,] *pc_pgm* [> *pc_pgm*]
`BLRS` `[`**`DB`**`,]` *`item_des`*

This command is available at PC level only.

Description: This command operates in the same way as the BL command but also resets the executions units (PCPGM, MASTER, CONTRM, SEQ). See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.13 C - Connect

Syntax: **C** *item_des* [:*terminal*]
`C` *`item_des`* `[:`*`function_parameter`*`]`
`C` `[`*`pc_pgm`*`,]` *`variable_name`*

This command is available at PC level only.

Description: Most of the elements, which have been chosen by insert statement, have inputs and outputs which can be connected in one of the following ways.

Input terminals:

- To an output terminal of another element
- To an input from the data base
- To a constant
- To an operational parameter.

Output terminals:

- To one or more element inputs
- To one or more outputs to the data base.

If a variable name but no PC program number was given the command assumes last PC program used.

A broken connection will produce one of the following results:

- If the source is an element output, the name and the unit are retained.
- If the source is constant or operational parameter it will cease to exist, along with its name and unit, when the last user is disconnected.
- If the source is a point in the data base, the data base remains unchanged, but the PC variable reflecting the value of the data base point will cease to exist; the “local” name and unit will also disappear.

The following subcommands are available:

!	Ends the dialog
>**G**	Switches from non-graphic to graphic representation
>**NG**	Switches from graphic to non-graphic representation
>**S**	Opens the Name/Unit dialog to assign a Name and Unit to a terminal
>	Goto next terminal
<	Goto previous terminal
=	Remain on same terminal
>>	Goto next element
<<	Goto previous element
>**:***terminal*	Jump to terminal within the element
>>*item_des*	Jump to first terminal of specified item designation
>>*item_des*[**:***terminal*]	Jump to terminal of specified item designation
>>*variable_name*	Jump to terminal named *variable_name*

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.14 CALCCHS - CALCulate CHeckSum

Syntax: **CALCCHS** [*start_addr* > *end_addr*][;**RWM**]

`CALCCHS [start_addr ; size][;RWM]`

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command calculates checksums for a given part of the system. The checksums are assumed to reside in PROM and a segment containing the new data is created. If the checksums resides in the RWM area the option “RWM” must be given and the new data is written directly in the target system. The segment can be input to a PROM programmer.

This command is not available with the Engineering Board PU516 and the RTA board.

Parameters The address range can be specified in two ways:

1. From “start_addr” to “end_addr” (Syntax 1).
2. From “start_addr” and “size” bytes (Syntax 2).

If no address range is given, new checksums are calculated for the whole at system generation specified area. Type HELP INSTALL for more information.

3.15 CIP - Connect InPort

Syntax: **CIP** *task_name,port_no=channel_name*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command connects an inport of a task to a channel. Type HELP TASK_NAME for description of task names. “port_no” is inport number (0 - 127). “channel_name” is name of channel (1 to 6 characters).

When an inport is connected, this task becomes the first consumer of signals on the concerned channel. The inport should have been disabled earlier, or not connected at all, in order to be connected. Connecting an inport does not mean that it is enabled. This must be done using command EIP.

Example: To connect inport 3 of task DEFS003 to channel SYSCHA:

```
CIP DEFS003,3=SYSCHA
```

3.16 CLLFILE - CLose LogFILE

Syntax: **CLLFILE**

This command is available at PC and system level.

Description: This command requests the system to stop logging of input/output (logging is started with “OLFILE”). See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.17 CLOSE - CLOSE segment

Syntax: **CLOSE** *volume*:*segment*

This command is available at system level only.

Description: Closes a segment opened previously via the command OPEN. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

This command is not available with the Engineering Board PU516 and the RTA board.

Example: To close a segment named AROW.SA on volume SOUR:

```
CLOSE SOUR:AROW.SA
```

3.18 COMPARE - COMPARE segments

Syntax: **COMPARE** [*volume_1*]:[*segment_1*]>[*volume_2*]:[*segment_2*]

This command is available at PC and system level.

Description: This command compares a segment on a volume with another segment on another or the same volume. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Example: To compare segment A.SA with B.SA both on volume AAAA:

```
COMPARE AAAA:A.SA>AAAA:B.SA
```

3.19 CONTINUE

Syntax: **CONTINUE**

This command is available at PC and system level.

Description: This command resumes the execution of a command file (which was interrupted by "USERINPUT"). See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.20 COP - Connect OutPort

Syntax: **COP** *task_name,port_no=channel_name*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: This command connects an outport of a task to a channel. Type HELP TASK_NAME for description of task names. "port_no" is outport number (0 - 127). "channel_name" is name of channel (1 to 6 characters). If the outport is connected already, it must be disconnected using command "DCOP" before connecting it to a new channel.

Example: To connect outport 2 of task DEFS003 to channel SYSCHA:

```
COP DEFS003,2=SYSCHA
```

3.21 COPY - COPY segment

Syntax: **COPY** [*volume_1*]:[*segment_1*]>[*volume_2*]:[*segment_2*][;**Y**]

This command is available at PC and system level.

Description: This command copies segment(s) within the same volume or from one to another.

“volume_1” is the source volume

“segment_1” is source segment(s) to be copied.

“volume_2” is the destination volume

“seg_2” is name(s) of destination segment(s).

If “;Y” is given in the command input, existing output segment(s) is(are) overwritten without confirmation. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Example 1: To copy segment CXAD:MYFILE01.SA to DXAP:YOURFILE.SA:

```
COPY CXAD:MYFILE01.SA>DXAP:YOURFILE.SA
```

Example 2: To copy all segments with extension RO:

```
COPY CXAD:*.RO>DXAP:*.RO
```

3.22 COSESS - Connect Offline SESSion

Syntax: **COSESS** [*nonsymbolic*]

Description: This command sets up an OFFLINE session to the physically connected target system, which is set in OFFLINE mode.

If parameter NOSYMBOLIC is given, then no data is retrieved from the target system (useful when analyzing a corrupt target system).

This command is not available with the Engineering Board DSPU131 and the RTA board.

3.23 CRDB - CReate Data Base

Syntax: **CRDB**

`CRDB` *`element-type`*

This command is available at PC and system level.

Description: This command enables addition of:

- One data base element,
- A number of elements that are related to each other. For example, one board element and its channel elements.

CRDB is an interactive command and will prompt the user for all required input, including the initial parameter (if omitted). See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.24 CREATE - CREATE segment

Syntax: **CREATE** [*volume*]:[*segment*],*type*,*size*[;**Q**]

This command is available at PC and system level.

Description: This command creates a segment on a volume. The segment is cleared, i.e. all data in the segment is set to 0. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name. “type” is C for Continuous and S for Sequential segments. “size” is size of the segment in blocks of 1 kbytes. If “;Q” is specified the segment is created “quick and dirty”, i.e. the segment is not cleared, data will be undefined.

Example: To create a continuous segment of size 120 Kbyte with name A.LO on default volume:

```
CREATE :A.LO,C,120
```

3.25 CSPT - CompreSs Program Tables

Syntax: **CSPT** *pc_pgm*

This command is available at PC level only.

Description: This command is redundant for certain system types. For those, the command terminates after a printout about this. The command is used to display the empty sizes in:

- A PC program
- The five PC program data tables
- The system.

and to compress the empty areas in the PC program and in its five PC program data tables.

Example: Call command with pc_pgm PC4:

```
#CSPT PC4
```

3.26 CSSD - CompresS System Data

Syntax: **CSSD**

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: This command compresses the system data area. The data is written to a segment on a floppy disk, which can be input to a PROM programmer. The target can only be an “off-line” system which has not been previously initiated. Type HELP INSTALL for more information. This command is not available with the Engineering Board PU516 and the RTA board.

3.27 CU - Connect Undefined

Syntax: **CU**

```
CU item_des [:terminal]
CU item_des [:function_parameter]
CU [pc_pgm,] variable_name
```

This command is available at PC level only.

Description: This command operates in the same way as the command Connect (C), but only processes those function parameters and input terminals without allocated values or connections. It can be useful to process the inputs listed as incomplete after attempting to Disable Build Mode (DIBM). See *Master Piece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: Disable build mode in PC3 and connect undefined terminals:

```
#DIBM PC3
UNDEFINED INPUTS:
PC3.1.4:1
#CU PC3.1.4:1
PC3.1.4
:1 =DI1.8
```

3.28 DAABF

Syntax: **DAABF [volume]:segment,name**

This command is available at PC and system level.

Description: volume:Destination volume; default is SRCE

segment:	Name of the segment to be dumped
name:	Name of the data base element from which the data should be dumped. Name can be either the element name or the element id

This command is used to dump an instance of data base element AAB_FILE to an AAB file on the harddisk. If the command is given without parameters, parameters are requested in a dialog.

Example 1: Dump data base element AAB_FILE1 to segment MYAAB1.DD.

```
#DAABF SRCE:MYAAB1.DD,AAB_FILE1  or
#DAABF MYAAB1.DD,AAB_FILE1
```

Example 2: Dump data base element with element name MY_AAB_FILE to segment MYAAB2.DD:

```
#DAABF SRCE:MYAAB2.DD,MY_AAB_FILE  or
#DAABF MYAAB2.DD,MY_AAB_FILE
```

3.29 DBL - DeBLock

Syntax: **DBL** *pc_pgm* [> *pc_pgm*]

`DBL item_des`

This command is available at PC level only.

Description: This command removes the blocked status from executing units (PCPGM, MASTER, CONTRM, SEQ) and places those units in the active status. See *MasterPiece200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Deblock program PC1:

```
#DBL PC1
```

Example 2: Deblock execution of programs PC2, PC3 and PC4:

```
#DBL PC2>PC4
```

3.30 DCOP - DisConnect OutPort

Syntax: **DCOP** *task_name,port_no*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command disconnects an outport of a task from the channel to which it is connected. Type HELP TASK_NAME for description of task names. “port_no” is outport number (0 ..127). Any signal sent by the task through this outport subsequently will be simply thrown away.

Example: To disconnect outport 2 of task DEFS003:

```
DCOP DEFS003,2
```

3.31 DCTARG - DisConnect TARGet

Syntax: **DCTARG**

Description: This command disconnects current target system, thus making it accessible for other users.

3.32 DD - Delete Debug point

Syntax: **DD** [*debug_no*] [,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command removes the specified debug points. “debug_no” is the debug point to be removed (0 - 7). If none are specified, all the defined debug points are deleted.

Example: To delete debug points 1,3,5:

```
DD 1,3,5
```

3.33 DDB - Delete Data Base

Syntax:

DDB
DDB *element-name*
DDB *item-des*
DDB *logic-file.logic-rec*
DDB *bus,station*[,*position*[,*subposition*]]

This command is available at PC and system level.

Description: This command enables deletion of:

- One data base element,
- A number of elements related to each other, for example, one board element and its channel elements.

DDB is an interactive command and will prompt the user for all required input, including the initial parameter (if omitted).

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.34 DDC - Delete Debug point Counter

Syntax: **DDC** [*debug_no*] [,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: This command removes the debug point counter selected via the command DFDC. “debug_no” is debug point for which counter is cleared. If none specified all the defined counters are cleared.

Example: To remove the counters for debug points 2,4,6:

```
DDC 2,4,6
```

3.35 DDCD - Delete Debug point Catch Data

Syntax: **DDCD** [*debug_no*] [,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command removes the data specified to be caught on passing specified debug point of Catch type. “debug_no” is associated debug point. If none specified all the defined Catch points are assumed.

Example: To delete debug point catch data for 2,6 and 7:

```
DDCD 2,6,7
```

3.36 DDCON - Delete Debug point CONdition

Syntax: **DDCON** [*debug_no*] [,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command removes the trigger condition (defined via DFDCON) for one or more or all debug points. “debug_no” is debug point for which the trigger condition is to be removed.
If none specified all debug points are assumed.

Example: To delete trigger condition set for debug points 1 and 5:

```
DDCON 1,5
```

3.37 DDTD - Delete Debug point Trace Data

Syntax: **DDTD**

WARNING
This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command removes the data units specified via command DFDTD to be printed on triggering the Trace point 0.

3.38 DDTSL - Delete Debug point control of Task Switch Log

Syntax: **DDTSL** [*debug_no*] [,*debug_no*]..[,*debug_no*]

WARNING
This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command removes the defined control of task switch logging for specified debug points. “debug_no” is the debug point in question. If none specified, all debug points are assumed.

Example: To delete control of task switch logging for debug point 3:

```
DDTSL 3
```

3.39 DELAY

Syntax: **DELAY** *delay-time*

This command is available at PC and system level.

Description: This command makes a time delay in steps of 0.1 seconds. See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.40 DELDOC

Syntax: **DELDOC**

This command is available at PC and system level.

Description: This command deletes all pages from the printer buffer without printing them.

See also: NEWDOC

3.41 DELETE - DELETE segment

Syntax: **DELETE** [*volume*]:*segment*[;**Y**]

This command is available at PC and system level.

Description: This command deletes one or more segments from a volume. If “volume” is omitted previously specified name is assumed. “segment” is segment(s) to be deleted. If omitted, previously specified name is assumed. Request for deletion is confirmed via a question for each segment in the segments list specified. If “;Y” is given in the command input, confirmation is not requested. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Please note that the command will only delete up to 100 files at a time. If your volume contains more files, enter the command a 2nd time to delete the remaining files.

Example: To delete the segment MY.RO on volume ZAXA:

```
DELETE ZAXA:MY.RO
```

3.42 DELEVENT - DELete EVENTs and alarms

Syntax: **DELEVENT**

This command is available at system level only.

Description: The DELEVENT command makes it possible to delete events and alarms in Advant Station 500 Series Operator Station. The command may be executed in both off-line and on-line session.

NOTE

Deleted unacknowledged alarms are still unacknowledged in process data base. You must be aware of that you will loose all incoming events and alarms during execution of DELEVENT. Presentation of all lists is blocked during execution of the command.

3.43 DFAD - DeFine Absolute Debug point

Syntax: **DFAD** *debug_no*,[*task_name*],*addr*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command defines a debug point at an absolute address. “debug_no” is debug point number (0 - 7). If task name is omitted, the last name entered is assumed. “addr” is address at which debug point is defined.

Example: To define debug point 1 at address H’AAB2 in task DEFS003:

```
DFAD 1,DEFS003,H’AAB2
```

3.44 DFDC - DeFine Debug point Counter

Syntax: **DFDC** *debug_no*,*value*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command defines a condition that a debug point should be passed given number of times before it is triggered. “debug_no” is debug point for which condition is set (0..7). “value” is number of times debug point should be passed (max 32767). Between conditions set by commands DFDC and DFDCON, Boolean OR applies.

Example: The debug point 3 will be passed 70 times before triggering applies:

```
DFDC 3,70
```

3.45 DFDCD - DeFine Debug point Catch Data

Syntax: **DFDCD** *debug_no*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command defines data units, which should be logged when a debug point of Catch type is passed. “debug_no” is associated debug point of type Catch. “addr_type” is type of address referring to data units. It can be SD (System Data), TC (Task Code), TDB (Task Data Base), TDC (Task Data Current) or REG (register). “data_type” is type of data to be logged. String is not allowed. Type HELP DATA_TYPE for description of data types. “addr” is the address of the data. Maximum number of data units allowed is 2. It is possible to terminate the input by entering “!”.

Example: To log Catch point 1:

```
DFDCD 1
```

Above command causes the display “DATA?”. To log an integer at base offset -8, after “DATA?” type:

```
DATA? TDB,SWD,-8
```

To log a real at base offset -2 as second data, type:

```
DATA? TDB,R,-2
```

3.46 DFDCON - DeFine Debug point CONdition

Syntax: **DFDCON** *debug_no*,*addr_type*,*data_type*,*addr*,*condition*,*val*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command makes a debug point conditional. The contents of a specified memory location is compared with a given value. If the Boolean result is TRUE, the debug point is triggered.

“debug_no” is the debug point to be triggered conditionally. “addr_type” is address type used to specify memory location to be compared with a given value. It can be TDB (Task Data Base), TDC (Task Data Current) etc. “data_type” is type of data to be assumed in the comparison. String is not allowed. Type HELP DATA_TYPE for description of data types. “addr” is the address of the data. “condition” is the type of comparison to be made in Motorola M68000 assembly mnemonic notation. “val” is value against which the comparison is to be made.

Example: To trigger debug point 3 when word at base offset -6 = 20:

```
DFDCON 3,TDB,SWD,-6,EQ,20
```

3.47 DFDT - DeFine Debug point Type

Syntax: **DFDT** *debug_no*,*debug_type*[/**P**][;*steps*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command defines function type for a debug point which has been defined already via command DFAD or DFRD. “debug_no” is debug point number (0 - 7). “debug_type” is type of debug point (B,C,T for Break, Catch, Trace). Type T is valid only for debug point 0. (/P) is only for relative debug points in a PASCAL task. “steps” is no of steps to be executed before breaking (Only for type Trace.)

Example 1: To define debug point 4 as break point in a PASCAL program:

```
DFDT 4,B/P
```

Example 2: To define a Trace point with step of 3 instructions:

```
DFDT 0,T;3
```

3.48 DFDTD - DeFine Debug point Trace Data

Syntax: **DFDTD** *addr_type*,*data_type*,*addr*[**!**]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: This command specifies data units which are to be printed when the Trace point 0 triggers. Maximum 10 data units can be printed. “addr_type” is address type used to specify memory location to be printed. It can be SD (System Data), TC (Task Code), TDB (Task Data Base), TDC (Task Data Current) or REG (register). “data_type” is type of data to be assumed for printing. Type HELP DATA_TYPE for description of data types. “addr” is the address of the data.

Example To define data units to be printed in Trace mode:

```
DFDTD
```

Now the system prompts with “DATA?”. To specify data units, enter one by one after “DATA?”:

```
DATA? REG,ULH,A2
DATA? TDB,UBD,-119
```

Finish with the terminator character (!) at the end.

3.49 DFDTSL - DeFine Debug point control of Task Switch Log

Syntax: **DFDTSL** *debug_no*,**ON**

```
DFDTSL debug_no,OFF
```

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command activates or deactivates logging of the task switches when the debug point triggers. “debug_no” is the debug point in question.

Example: To activate task switch logging for debug point 4:

```
DFDTSL 4,ON
```

3.50 DFLSYMB - DeFine Local SYMBol

Syntax: **DFLSYMB** *symbol=expression*

This command is available at PC and system level.

Description: This command creates a local symbol in current command file levels’ symbol table. The symbol is redefined if it exists.

symbol Name of symbol

expression Defines the value of the symbol

See Section 6.5.3, Defining Symbols, for more information.

Examples (4):

```
DFLSYMB base = H’1C1A
DFLSYMB level = 1.25
DFLSYMB volume = "VOLA"
DFLSYMB X = F$EXTRACT(1,F$LENGTH(A),A)
```

In example 1 a symbol with a hexadecimal value is created. The value in example 2 is a real number. Example 3 shows a symbol with a text value associated to it. Example 4 shows how to use a lexical function.

3.51 DFNETUSE - DeFine NETwork USE

Syntax: **DFNETUSE** [**ON**|**OFF**]

WARNING

This command shall be used by experts only!

This command is not available with the Engineering Board DSPU131 and the RTA board. For PU516, the command is available at PC and system level.

Description: This command enables (option ON) or disables (option OFF) the target system connection via MB300. If no parameters are specified, the command displays the current communication settings.

Example To show the current communication settings enter:

```
DFNETUSE
```

3.52 DFRD - DeFine Relative Debug point

Syntax: **DFRD** *debug_no*,[*task_name*],*addr*

```
DFRD debug_no,[task_name],statement
```

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description:

This command defines a relative debug point either at an address or at a statement in a task. “debug_no” is debug point number (0 - 7). If task name is omitted latest entered name is assumed. “addr” is offset of debug point from the start of the task code. “statement” is debug point statement number (Only for PASCAL programs).

Example 1:

To define debug point 4 at offset H’AAB2 in task DEFS003:

```
DFRD 4,DEFS003,H’AAB2
```

Example 2:

To define debug point 6 at statement no 167 in task DEFS003:

```
DFRD 6,DEFS003,167
```

3.53 DFSYMB - DeFine (global) SYMBol

Syntax: **DFSYMB** *symbol* = *expression*

This command is available at PC and system level.

Description: This command creates a global symbol. The symbol is redefined if it exists.

symbol	Name of symbol
expression	Defines the value of the symbol

See Section 6.5.3, Defining Symbols, for more information.

Examples (4):

```
DFSYMB PC = 10
DFSYMB ITEM = "PC''PC'.2"
DFSYMB X = (1.2E2/3.4)-(-19*('Y'+1.1))
DFSYMB X = F$TYPE(A)
```

In example 1 symbol PC is assigned decimal integer value 10. In example 2 symbol ITEM is assigned value “PC10.2” using a text expression involving the previously defined symbol PC. In example 3 X is assigned a floating point value through an arithmetic expression, also involving another symbol Y. Example 4 shows how to use lexical functions as expressions.

3.54 DIBM - DIsable Build Mode

Syntax: **DIBM** *pc_pgm*[>*pc_pgm*]

This command is available at PC level only.

Description: The **DIBM** (DIsable Build Mode) command is used to end building of one or more programs. The program(s) are blocked afterwards and can be changed to the execution mode using the **DBL** command.

This command resets all data base values that are connected to the output terminals of the involved PC program(s) to zero.

If non-connected inputs exist, they are listed and the build mode is not concluded. Connect those unconnected inputs and repeat the **DIBM** command. See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Disable Build Mode in PC8 with terminal error:

```
#DIBM PC8
UNDEFINED INPUTS:
PC 8.1.1:1
```

(where:1 is the unconnected terminal on element 8.1.1)

Example 2: Disable Build Mode in PC5, PC6, PC7:

```
#DIBM PC5>PC7
```

3.55 DICONFIG - DIsable CONFIGuration

Syntax: **DICONFIG**

This command is available at PC and system level.

Description: This command sets the system in operation mode. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.56 DID - DIsable Debug point

Syntax: **DID** [*debug_no*][,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command disables one or several debug points, enabled via command ED. “debug_no” is debug point number (0 - 7). If no debug point is specified, all the enabled debug points are disabled.

Example: To disable debug point 5,2,6:

```
DID 5,2,6
```

3.57 DIIP - DIsable InPort

Syntax: **DIIP** *task_name,port_no*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command disables an inport of a task. Type HELP TASK_NAME for description of task names. “port_no” is inport number (0 - 127).
When an inport is disabled, any signal on the channel is passed on to the next task in the receiver chain. A receive operation from the task, whose inport has been disabled, returns a NIL pointer. A wait and receive operation causes the task to be suspended until the inport is enabled (and signal is available) or time out occurs.

Example: To disable inport 3 of task DEFS003:

```
DIIP DEFS003,3
```

3.58 DIMDB - DIMension Data Base

Syntax: **DIMDB**

This command is available at PC level only.

Description: This command is used to dimension the data base in a target system.

The command works in split screen mode. The upper part of the screen shows the present data base dimensions and the lower part holds the dialog scratch pad.

Subcommands are **<Enter>** or one of **S**, **S0 - S**6, **M**, **Mz**, **MLz, >**, <, **=**, **DIM**, **END**, or **Q** followed by **<Enter>**. In **Mz** or **MLz**, z is the logical file number. Press **<Ctrl-Break>** to break.

Basic Mode At entry of the command the S0 picture (section names only) is shown. One data base section at a time is pointed out on the screen by the > or < sign. The logic points in the pointed out section can be inspected or modified by the subcommands S, Sx (x = section number) or M. The M subcommand sets the command in modify mode.

Modify Mode The command is set in Modify mode by the M, Mz or MLz subcommands (z = logical file number). The subcommand ! (exclamation mark) resets the command in basic mode.

The concepts of the open section are indicated, one at a time, by the > or < sign. The name, the number of the basic logical file and the current number of instances are shown for all concepts in the open section. Name and number of instances are also shown at the bottom of the screen.

The operator has to modify the number or approve it, and just hit the **<Enter>** key. A modified value is denoted by an * at the left.

At completion of one section, the next section is automatically opened, with the indicator at the first concept.

Termination of the command

DIM is the normal termination subcommand. It defines the data base spare area, to be used for eventual later expansion of any concept, and erects the data base according to the entered dimensions.

The subcommand END is used if you regret and do not want to dimension or, within the LOTDB command, redimension.

NOTE

END within the LOTDB has same effect as LOTDB without the redimension activity.

The subcommand QUIT or Q cancels the command. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.59 DIMP - DIsable Modify Permission

Syntax: **DIMP** *item_des*[>*item_des*]

This command is available at PC level only.

Description: This command deactivates the modification permission for one or more execution units (PCPGM, CONTRM, MASTER, SEQ). If some execution unit concerned does not satisfy the requirements for deactivation, the command is not executed and a message indicates which execution unit has not satisfied the requirement. The two item designations must refer to a PC program or an execution unit. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Disable modify permission in PC2 and PC3 with terminal error:

```
#DIMP PC2>PC3
```

Example 2: BUILD MODE OR BLOCKINGS IN

PC3.4

```
#DBL PC3.4
#DIMP PC2>PC3
```

3.60 DIMPC - DIMension PC

Syntax: **DIMPC**

This command is available at PC level only.

Description: This command is used to display the attributes of the current system, and allows the user to redimension any or all of the variable attributes.

There are a number of subcommands available in **DIMPC**:

M	The line at the cursor (>) is opened for modification.
!	Subcommand “**!**” is used to switch from the modify mode to the initial mode.
DIM	The requested dimensioning is transferred to the Advant Controller 400 Series with the subcommand **DIM**.
PRINT	The subcommand **PRINT** gives a printout of the screen contents.
END	The **END** subcommand is used to end **DIMPC**.
MW*x*	Move Window, whereas $-99 \le x \le 99$.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.61 DINS - Delete INStance

Syntax: **DINS** *logic_file*,*logic_rec*

`DINS logic_file,logic_rec;key`

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command deletes an instance from the data base. The instance to delete can be defined explicitly or indirectly by using a key. Type HELP KEY for the description of key condition. Confirmation of the deletion is made by answering the question:

```
DELETE (Y/N)?
```

Example: Delete record 2 of logical file 5

```
DINS 5,2
```

3.62 DITO - DIsable Time Out

Syntax: **DITO**

WARNING

This command shall be used by experts only!

This command is available at PC and system level.

Description: This command deactivates diagnostic time out generation.

3.63 DMEMB - Delete MEMBer

Syntax: **DMEMB** *mlevref*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command deletes a member defined in the property. If no member number is defined with the property, or the number is greater than the last current number then the last member in the group will be deleted. Type HELP MLEVREF for the description of mlevref parameter.

3.64 DS - Delete Statement

Syntax: **DS** *item_des*[>*single_level_item_des*]

This command is available at PC level only.

Description: The DS command can be used to delete a single or a group of statements. The start and end item designations must be on the same level in the program structure; any intermediate item designations must also be on this, or a lower, level.

Name and unit information for a item designation’s terminal is deleted as well as the instance name assigned to it.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Delete one statement, PC1.1.5

```
#DS PC1.1.5
```

Example 2: Delete a group of statements

```
#DS PC1.1.1>.4
```

Example 3: Delete structure element PC1.1 and all subordinate elements.

```
#DS PC1.1
```

3.65 DSYMB - Delete (global) SYMBol

Syntax: **DSYMB** [*symbol*][;**Y**]

This command is available at PC and system level.

Description: This command deletes the specified global symbol. If the name is omitted, selective deletion of all the global symbols is assumed. If there is no global symbol with the given name, selective deletion of all the global symbols with names starting with the given name is assumed. Selective deletion is done in query mode, unless “;Y” is given.

symbol: name of symbol

See also help on keyword SYMBOLS. See Section 6.5.5, Deleting Symbols, for more information.

Examples:

```
DSYMB PC_1_1
DSYMB PC_1
DSYMB
```

In example 1 global symbol PC_1_1 is deleted. Example 2 shows how PC_1_1 and possibly some or all other symbols with names starting with PC_1 can be deleted in query mode. Example 3 shows how some or all the symbols in the symbol table can be deleted in query mode.

3.66 DUAP - DUmp APplication

Syntax: **DUAP** *dump_name*[,vol*ume*][;*options*]

where

- dump_name = 1 to 7 alphanumeric characters
- options = a string of one or more option identifiers
- volume = SRCE or USER.

This command is available at PC and system level.

Description: The command DUAP (DUmp APplication) is used to create a backup dump of an Advant Controller 400 Series. The DUAP dump contains all data base and PC applications present in the target. The command needs remote or off-line session.

The dump consists of 1 up to 16 segments and can have 257, 513, 1025 etc. kbytes. The last segment may be smaller or larger than the standard sized segment.

The command uses the default volume SRCE and can be used in service or in configuration mode. If possible, configuration mode should be used, since the time required for dumping will be shorter.

The segment names of the dump contain the given name, one or more zeros and a sequential number (corresponding to the order between segments). The type (extension) of the segments is AD.

You can enter a dump description (one single line, max. 80 characters) and a message to the person who loads the dump later on (max. 8 lines of 80 characters). The dump messages can be used for information about the dump, which system is belongs to etc.

Option identifiers: **D**:(Delete) all existing segments having the same names as the new dump segments are going to get, are automatically deleted from the volumes. By default, in dialog mode deletion needs operator’s permission. In parameter mode the command terminates if a duplicate segment is found.

F:	Forces segment size to be 257 kbytes.
I:	(Information) gives more progress text, than by default.
N:	(No questions) suppresses the prompting for dump message texts. By default, the operator is prompted for two text messages. In both dialog and parameter modes.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: If you want to know what size the dump is going to get, you just enter “**DUAP X**”. The size will be estimated and shown on the screen along with a “VOLUME” question. Terminate the command by entering “Q”.

Example 2: Dump command in dialog mode, no options:

```
DUAP MYDUMP
```

An application dump named MYDUMP is created on the SRCE volume. with a block, containg dump information. Note, that the information block may be listed by use of the LIST command.

Example 3: Dump command in parameter mode with D, I and N-options:

```
DUAP MYDUMP,SRCE;DIN
```

With a total estimated dump size of 800 kbyte and a volume, which has enough of free space, two dump segments are created:

SRCE:MYDUMP01.AD of 513 kbyte and SRCE:MYDUMP02.AD of 289 kbyte.

3.67 DUCON - DUmp CONcept(s)

Syntax: **DUCON**
DUCON *volume*:*segment*/**C**=*concept*[,...]
DUCON *volume*:*segment*/**L**=*logical file*[,...]

WARNING

This command shall be used by experts only!

This command is available at PC and system level.

Description: volume: Volume name, 4 characters

segment: Segment name, max. 6 characters, no extension

concept: Concept(s) to be dumped; * = dump all

logical file: Logical file(s) to be dumped; * = dump all

If this command is used without parameters, it starts a dialog, in which the user can specify the name of the dump volume and select concepts from the target system data base.

If the command is used with parameters, the selected concept(s)/logical file(s) will be dumped to segment:

NAMEnn.CD (nn = 01 - 99) on the given volume.

Example 1: Dump the concepts AI_INTERFACE and AO_INTERFACE to the segment KLI on volume SRCE.

```
*DUCON SRCE:KLI/C=AI_INTERFACE,AO_INTERFACE
```

The segment(s) KLI01.CD ... KLInn.CD are created on volume SRCE.

Example 2: Dump the logical files 300,301,302 to the segment TABLE on volume SRCE

```
*DUCON SRCE:TABLE/L=300,301,302
```

The segment(s) TABLE01.CD ... TABLEnn.CD are created on volume SRCE.

Example 3: Dump all concepts in the system to the segment ALL on volume SRCE

```
*DUCON SRCE:ALL/C=*
```

The segment(s) ALL01.CD ... ALLnn.CD are created on volume SRCE.

3.68 DUDB - DUmp Data Base

Syntax: **DUDB** *volume* : *segment*

`DUDB volume : segment ; logic_file`

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command can be used to create a logical dump. If the third parameter is not given, you will be asked to enter it. Type HELP VOLUME_NAME for description of volume. Type HELP SEGMENT_NAME for description of segment. See documentation of *Advant Station 500 Series Operator Station User's Guide* or *MasterView 800/1* for more information.

3.69 DUDBS - DUmp Data Base Source

Syntax: **DUDBS** [*volume*:][*segment*][**;L**][,**SEL**=*selection*][*Opt_param*]

This command is available at PC and system level.

Description: This command can be used to dump and/or print data base source code. If necessary arguments are missing, the command switches to an interactive mode.

;**L**	Prints the data base source code.

The argument “selection” specifies the elements to dump:

- All data base elements in every category, (SEL=ALL).
- Selection of data base elements using wildcards, (e.g. SEL=A*, which selects all data base elements starting with “A”); the wildcard “*” is allowed at the end of the selection string only.
- All data base elements of one type, (e.g. SEL=DIC).
- A range of elements of the same type, (e.g. SEL=AI1>4).
- One specific data base element, (e.g. SEL=AIC6).
- One specific data base element referenced by its name, (e.g. SEL=XYZ).
- A range of elements of the same type via their logical file/logical record number, (e.g. SEL=334.1>5).
- One specific data base element via its logical file/logical record number (e.g. SEL=334.1).

The argument “Opt_param” is as follows (single, or in combination):

/**NOGD**	No general defaults included.
/**NOSYS**	No system data base elements included
/**FULL**	Defaults included for each element.
=*maxsize*	Maximum segment size, default = 1024 blocks.
/**OUTPUT**	Output values also dumped.

/**MODIFY**	A "MODIFY" keyword precedes the element name(s) in the dump file.
/**TBL**	Table data included for element TBL_PARAM.

If no "Opt_param" is specified, the general defaults are dumped ONCE for each element type and then for each element ONLY the input values differing from the defaults.

NOTES

- The first element in the range MUST exist. In Example 3, DIC1 must exist.
- A single subordinate element cannot be dumped. For example, if you wish to dump AI3.4, the elements AI3 and all its related subordinate elements AI3.1 - AI3.16 must also be dumped.
- Source code can be reloaded and translated by the command: **TRDBS** - TRanslate Data Base Source.

Example 1: Dump all DI Calculated elements to disk only:

```
#DUDBS SRCE:DABASE,SEL=DIC
```

Example 2: Print the data base source code for all data base elements, omitting the leading default values section of the listing:

```
#DUDBS ,SEL=ALL/NOGD;L
```

Example 3: Dump the input and output values from the first three elements of type DI Calculated and print the source code:

```
#DUDBS SRCE:DABASE,SEL=DIC1>3/OUTPUT;L
```

Example 4: Dump the first element of DI Calculated with a "MODIFY" keyword preceding the element name:

```
#DUDBS SRCE:DABASE,SEL=DIC1/MODIFY
```

Example 5: Use of wildcards within the selection:

```
#DUDBS SRCE:DABASE,SEL=A*
```

This command dumps all data base element instances starting with "A", e.g. AI, AO, AIPT, AIC

3.70 DUMPFILE

Syntax: **DUMPFILE** *name*[,*filename*]

where

nameName of the data base element of type **AAB_FILE** which shall be dumped.

filenameVolume:segment.extension

This command is available at PC and system level.

The **DUMPFILE** command dumps the contents of data base element *name* of type **AAB_FILE** into the segment *filename*. This command is needed when **User Defined PC Elements** shall be dumped from Advant Controller 400 Series. For more information see *User Defined PC Elements User's Guide*.

3.71 DUPCS - DUmp PC Source

Syntax: **DUPCS** *item_des* [>*item_des*]/[*volume*:]*segment*[=*size*][;[**L**][**D**]]
DUPCS *item_des* [;*numb_elements*]/[*volume*:]*segment*[=*size*][;[**L**][**D**]]
DUPCS *item_des* [>*item_des*];**L**[**D**]
DUPCS *item_des* [;*numb_elements*];**L**[**D**]
This command is available at PC level only.

Description: This command is used to dump PC program(s) from target system to source code format on mass memory.

L	The source code dump is sent to the printer.
D	All PC scan table position parameters are omitted in the dump. Default volume name is SRCE.

If no volume and segment names are given, but an L parameter, the dump is listed on the printer only. No more than 4 characters may be given for the segment name. If the command is given without the parameter “size”, the size is 1024 blocks. See *ABB Master Source Code Handling* manual for more information.

3.72 DURWM - DUmp RWM

Syntax: **DURWM** *volume*:*segment*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command takes a backup of a system’s RWM data area. The backup is stored on one or several volumes, depending on the size of the dump, and starting with the segment given in the command. On each volume two segments are created: the first holds information about the dump (extension XI) and the second contains the data (extension XD).

Type HELP VOLUME_NAME for description of volume name. Segment is 1-7 characters, the eight being used for numbering the segments. Note that the extension should not be given. The command can be aborted pressing **<Ctrl-Break>**.

This command is not available with the Engineering Board PU516 and the RTA board.

Example: **DURWM VOLA:BACKUPA**

The dump will start with volume VOLA. Segment BACKUPAI.XI will hold information about the dump and BACKUPA1.XD the first part of data.

3.73 DUTDB - DUmp Total Data Base

Syntax: **DUTDB** [*dump_name*]

This command is available at PC and system level.

Description: dump_name1...6 characters

This command is used to create a total dump of the data base in the target. All data base concepts within the target system, are included in the dump. The command dialog is self instructive and the valid answers are listed within the prompting text of every question. The command can be terminated by pressing **<Ctrl-Break>** or, during the dialog, by entering the answer “Q” to any question. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.74 DUTPT - DUmp Total Program Tables

Syntax: **DUTPT** [*dump_name*]

`DUTPT [dump_name,volume]`

This command is available at PC level only.

Description: dump_name Name of the dump, 6 characters

volume Volume name, 4 characters

This command is used to dump the PC programs to the mass memory. The command operates in dialog mode. It compresses the PC area automatically before the dump takes place (Advant Controller 400 Series and MasterPiece 200/1). For other types of systems: use the CSPT (CompreSs Program Tables) command before dumping. Enter **<Ctrl-Break>** to break.

NOTE

> All blocked elements in the PC program are automatically deblocked when the program is loaded back into the system

The command will store the dump on the volume SRCE by default. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.75 DV - Delete Values

Syntax: **DV** [*table_name*]

This command is available at PC level only.

Description: This command is used to delete tables of values used by the command

. (NOTE: to delete a single value, the **DV** command must be called as a subcommand in GETAB). Two different types of tables exist:

- the default table, which does not have a name, and
- up to four named tables.

The DV command without the table_name parameter deletes the contents of the default table. The table itself cannot be deleted. If the table_name parameter is specified, the specified table will be deleted.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Examples: Delete the contents of the default table of values:

```
#DV
OK TO DELETE ENTIRE TABLE? (Y/N)Y
```

Delete the table Table_one completely:

```
#DV Table_one
OK TO DELETE ENTIRE TABLE? (Y/N)Y
```

3.76 EBM - Enable Build Mode

Syntax: **EBM** *pc_pgm*[>*pc_pgm*]
This command is available at PC level only.

Description: This command interrupts the execution of one or several PC programs. It makes program modifications possible without interference with the process control. Several non-consecutive PC programs require separate commands.

This command resets all data base values that are connected to the output terminals of the involved PC program(s) to zero.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Enable build mode in PC14:

```
#EBM PC14
```

Example 2: Enable build mode in PC18, PC19, PC20, PC21:

```
#EBM PC18>PC21
```

3.77 ECHO

Syntax: **ECHO ON|OFF**

This command is available at PC and system level.

Description: This command turns ON|OFF echoing of user command output. “ECHO” is only valid if a log file is open. See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.78 ECONFIG - Enable CONFIGuration

Syntax: **ECONFIG**

This command is available at PC and system level.

Description: This command sets the target system in configuration mode.See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.79 ED - Enable Debug point

Syntax: **ED** [*debug_no*][,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command enables one or several debug points, which has been defined (DFAD or DFRD) as well as type defined (DFDT). “debug_no” is debug point number (0 - 7). If no debug point is specified, all the defined debug points are enabled.

Example: To enable debug points 5,0,7:

```
ED 5,0,7
```

3.80 EIP - Enable InPort

Syntax: **EIP** *task_name,port_no*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command enables the inport of a task which had been disabled earlier. Type HELP TASK_NAME for description of task names. “port_no” is inport number (0 - 127).

Example: To enable inport 3 of task DEFS003:

```
EIP DEFS003,3
```

3.81 EMP - Enable Modify Permission

Syntax: **EMP** *item_des*[>*item_des*]

This command is available at PC level only.

Description: This command activates the modification permission for one or more execution units. The two item designations must refer to a PC program or an execution unit (PCPGM, CONTRM, MASTER, SEQ). See *Master Piece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Call command with PC programs PC3, PC4, PC5, PC6:

```
#EMP PC3>PC6
```

Example 2: Call command with execution unit PC3.5:

```
#EMP PC3.5
```

3.82 ETO - Enable Time Out

Syntax: **ETO**

WARNING

This command shall be used by experts only!

This command is available at PC and system level.

Description: This command activates diagnostic time out generation.

3.83 ETTY - Enable TTY mode

Syntax: **ETTY**
ETTY *target*
ETTY *target*:*baud*
ETTY *target*:*baud*;*echo*,*word*,*parity*,*stop*,*handshake*
This command is available at system level only.

Description: This command causes the engineering station to act like a VT100 terminal connected to a target computer. To exit from the terminal mode, press **<SHIFT-F4>**.

This command is not available with the Engineering Board PU516 and the RTA board.

- “target” is target computer (MA7 or DS101).
- “baud” is baud rate (110,300,600,1200,2400,4800,9600,19200).
- “echo” is local echo control (ECHO or NOEC; normal = ECHO.)
- “word” is bits per character (5 - 8).
- “parity” is parity type (N,O,E,1,0).
- “stop” is no of stop bits (1 or 2).

- “handshake” is transmission handshake (XON or OFF).

Example 1: To connect as a VT100 to MA700:

```
ETTY MA7
```

Default parameters: 4800;ECHO,8,N,1,XON

Example 2: To connect as a VT100 to an Advant Controller 400 Series:

```
ETTY DS101
```

Default parameters: 9600;ECHO,8,N,1,OFF

3.84 EXECFILE - EXEcute Command FILE

Syntax: **EXECFILE** *commandfile*[;*parameters*]
This command is available at PC and system level.

Description: This command passes execution control to the specified command file. All nested subroutine calls are skipped. “Parameters” are from 1 to 9 optional parameters to pass to the command file. See Section 6.2.1, EXECFILE - EXEcute Command FILE, for more information.

3.85 EXECSUB - EXEcute SUBroutine

Syntax: **EXECSUB** *commandfile*[;*parameters*]
This command is available at PC and system level.

Description: This command makes a subroutine call of the specified command file. Control is passed back to the calling file with “RETURN”. You can pass 1 up to 9 parameters to the subroutine. See Section 6.2.2, EXECSUB - EXEcute SUBroutine, for more information.

3.86 EXIT

Syntax: **EXIT**
This command is available at PC and system level.

Description: This command passes execution control back to the command level. All nested subroutine calls are skipped. See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.87 FDISP - Find DISPlay(s)

Syntax: **FDISP** *P1*[/*P2*][:*P3*][;*P4*]

This command is available at system level only.

Description: Lists all displays which includes a given object. See documentation *Advant Station 500 Series Operator Station User’s Guide or MasterView 800/1* for more information.

Parameters

P1:	Name or logical file and logical instance no of object.
P2:	Display name. Searching will start with this display. Default is first display.

P3:	Options (see below).
B (brief):	Lists object name, network node reference type and status.
L (long):	Lists object and display data plus logical file and logical instance.
P4:	Number of objects to search for. Default is 1.

Example 1: **FDISP 107,4**

Lists object and display data on logical file 107 and logical instance 4.

Example 2: **FDISP AI.CHANNEL.1/PROC.01:L**

Lists the occurrence of AI.CHANNEL.1 in all displays starting with PROC.01 in long mode.

Example 3: **FDISP 107,4:B;2**

Lists object data of instances no 4 and 5 in brief mode.

3.88 FINS - Find INStance

Syntax: **FINS** *logic_file*,*logic_rec*
FINS *logic_file*,*logic_rec*;*key*
This command is available at system level only.

Description: This command opens the specified instance for the commands GPRO, GINS, GFOR, MPRO, MINS, MFOR, FNEX, AMEMB, DMEMB. If given, a key shall control the access and the specified instance indicates the search start only. Type HELP KEY for the description of key condition.

Example: Find record 1 of logical file 4

```
*FINS 4,1
```

3.89 FNEX - Find NEXt Instance

Syntax: **FNEX**
This command is available at system level only.

Description: This command opens the next logical instance according to previous given keys. If no search conditions exists, then the next sequential instance will be opened. FNEX is only working on the current level.

3.90 GEPCD - GEnerate PC Diagram

Syntax: **GEPCD** [*item_des*]
GEPCD D [, *item_des*]
This command is available at PC level only.

Description: This command is used to display the current values of inputs and outputs of a PC element. The values can be both statically and dynamically updated and diagnosed, and the connections can be traced backwards. If a D is entered as a parameter, all values are inserted and displayed with dynamic updating. Use **<Ctrl-Break>** to get back to the command level with dynamically updating subcommands. The following subcommands are available for **GEPCD**:

HELP, **?**, **IV**, **GVD**, **MV**, **DV**, **GS**, **DIAG**, **GEPCD**, **MW**, **PRINT**, **RMD**, **!**, **END**

Subcommands:

HELP or **?**	displays available subcommands
IV	(Insert Value) inserts the current values of a selected terminal or variable on the display Syntax: `IV [variable_identity]`
GVD	(Get Values Dynamically) displays the status of variables dynamically Syntax: `GVD`
MV	(Modify Value) enables the modification of operational parameter and in some cases call parameter values. Syntax: `MV variable_identity` syntax: `MV item_des : call parameter`
DV	(Delete Value) deletes the current values of all or selected terminals on the display Syntax: `DV [variable_identity]`
GS	(Get Source) displays the source terminals for the selected input terminal Syntax: `GS terminal`
DIAG	(DIAGnose) traces back the source and diagnoses the cause of a particular output terminal value. PC elements displayed or those, which have more than one input (they can cause the status of the specified connection) and those which are connected to the data base. Syntax: `DIAG terminal`
GEPCD	(GEnerate PC Diagram) displays the PC element selected. If a "**D**" is entered as a parameter, values are inserted and displayed with dynamic updating. Syntax: `GEPCD D, item_des`
MW*x*	Moves the Window, whereas -99≤*x* ≤99.
PRINT	Outputs a copy of the current display on the printer Syntax: `PRINT`
RMD	(ReMove Display) terminates **GEPCD** command and removes display from the screen. If you start **GEPCD** again, the system starts from the PC element that was displayed before **RMD**. If you have terminated the **GEPCD** command with the subcommand "**!**", the PC element on the screen can be removed with the **RMD** (ReMove Display) command.
!	Terminates the command and leaves the display on the screen, but the variable values are not updated.
END	Terminates **GEPCD** command and removes display from screen

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.91 GEPD - GEnerate Page Division

Syntax: **GEPD** *pc_pgm [Opt_param]*

This command is available at PC level only.

Description: This command is used for automatic page division of a PC program. It generates hard page breaks.The argument Opt_param is as follows:

/D	All existing page headlines are deleted

Example 1: Generate page division for PC2

```
#GEPD PC2
```

Example 2: Generate page division for PC2 and delete the headlines

```
#GEPD PC2 /D
```

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.92 GETAB - GEnerate TABle of values

Syntax: **GETAB** [*table_name*][**/LT**]
where table_name can consist of 1 to 20 characters.
This command is available at PC level only.

Description: This command is used to display the current values of signals in PC programs and the data base. The values can be both statically and dynamically updated. With dynamically updating subcommands use **<Ctrl-Break>** to get back to the command level. The following data types can be presented: B, IB, I, IL, R, T, TR. Two different types of tables exist:

- The default table, which does not have a name, and
- Up to four named tables.

The default table is always present and cannot be deleted. When **GETAB** is started without a parameter, it will display the default table. If you specifiy the table_name parameter, GETAB will display that table. If the table does not exist, GETAB will create it.

GETAB /LT will list the currently available named tables.

Values can be displayed either in a table or in a graphical diagram. After start-up of the GETAB command you shall answer the following question:

```
Do you want graphical mode (G) or text mode (T)?
```

Type **G** for graphical or **T** for text mode.

The following subcommands are available for **GETAB**:

GVD, **PRINT**, **HELP**, **?**, **MV**, **IV**, **DV**, **RMD**, **!**, **END**

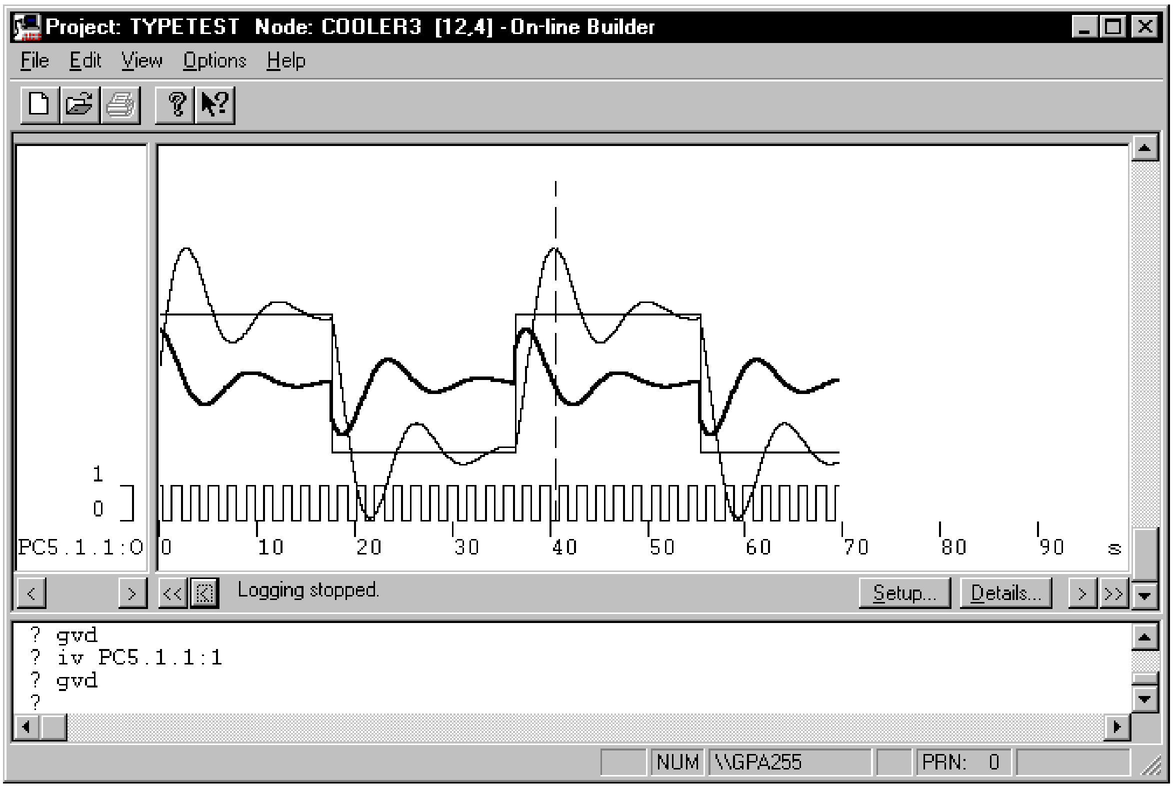
Subcommands:

GVD (Get Values Dynamically) displays the signal values in the table or in the diagram dynamically. In graphical mode (diagram) you can combine the recording of data with a trigger condition. Recording can be stopped with the

	<**Ctrl-Break**> command. If you start afterwards the recording again, the old data will be erased. Syntax: **`GVD`**
PRINT	Outputs a copy of the current display on the printer. This command is not available in graphical mode. syntax: **`PRINT`**
HELP or **?**	Displays available subcommands
MV	(Modify Value) enables the modification of operational parameter and in some cases call parameter values. The **MV** subcommand starts with a dialog Syntax: **`MV`** *`variable_identity`* Syntax: **`MV`** *`item_des`* `:` *`call parameter`*
IV	(Insert Value) inserts the current values of a selected terminal on the display Syntax: **`IV`** *`variable_identity`* Syntax: **`IV`** `[`**`DB`**`] =`*`data_base_identity`*
DV	(Delete Value) deletes the current values of all or selected terminals on the display of the selected table Syntax: **`DV`** `[`*`variable_identity`*`]` Syntax: **`DV`** `[`**`DB`**`] =`*`data_base_identity`*
RMD	(ReMove Display) terminates **GETAB** command and removes display from the screen
SAVE	Saves the contents of the current table in a file with default extension GLX. Values are separated by tabulators. The file can be opened with **AS100-EDIT** or with a spread sheet program. You can change the manner the data are displayed in AS100-EDIT with the same buttons as in the GETAB command. Limitation: AS100-EDIT is available on Advant Station 100 Series Engineering Stations, only. The SAVE command is available in graphical mode, only.
!	Terminates the command and leaves the display on the screen, no updates of the variables take place
END	Terminates **GETAB** command and removes display from screen.

The signal names you have entered with the **IV**(Insert Value) subcommand will be stored when **GETAB** terminates and are available at the next startup in both modes, text and graphical mode.

If you have chosen the graphical mode, you can setup the display to fit your needs. In Figure 3-1 four selected values are displayed.



```
? gvd
? iv PC5.1.1:1
? gvd
?
```

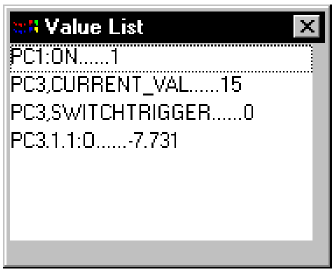
Figure 3-1. Display of GETAB in graphical mode

The left part of the display shows the vertical scale. For each displayed value an own scale is prepared, but for clarity only one scale is shown at a time. You can switch between different scales with the < or > buttons below the scale. In Figure 3-1 the scale for the binary signal PC1,SWITCH is selected. This scale an offset of one is assigned to. With an offset you can adjust different signals so that they do not overlap each other.

At the right side of the vertical scale the signals are displayed together with a common horizontal time scale. You can scroll within this area with the <<, <, > and >> buttons. With < and > the display will be scrolled by one unit, while << and >> scroll ten units when pressed.

Two other buttons are at the bottom line of the display: **Setup** and **Details**.

With **Setup** you can configure the display.The configuration is saved when leaving **GETAB**. Pressing the **Details** button will open another window called **Value List**, that shows a list of all values activated with the **IV** (Insert Value) subcommand. This window will show either the newest values or the values at the marker position. The marker - when positioned - is presented as a vertical, broken line as shown in Figure 3-1. You can set the marker by clicking at the desired position into the display. The **Value List** is shown in Figure 3-2.

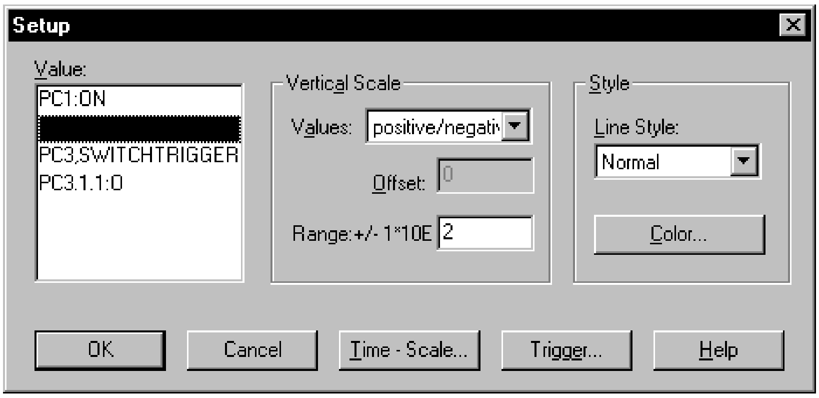


Value List

PC1:ON......1
PC3,CURRENT_VAL......15
PC3,SWITCHTRIGGER......0
PC3.1.1:0......-7.731

Figure 3-2. Value List window

The **Setup** window is shown in Figure 3-3.



Setup

Value:
PC1:ON
PC3,SWITCHTRIGGER
PC3.1.1:0

Vertical Scale
Values: positive/negati
Offset: 0
Range:+/- 1*10E 2

Style
Line Style: Normal

Figure 3-3. Setup window

Within the **Setup** you can specify for each value listed in the **Value** list box:

- the arrangement of the vertical scale
- the style and color of the displayed curve and the corresponding text.

The time scale you can adjust once for all signals. Further on you can define a common trigger condition for all signals.

Dialog Controls of Setup

Value

A list box with all signals activated with the **IV** (Insert Value) subcommand. Select one value and specify Range and Style for it. Repeat this procedure for all values listed.

Further on you can define a **Trigger** condition common for all signals.

Vertical Scale

Values:
Defines the principal spread of the vertical axis. Select one of the possible values:

- Positive
  only positive values will be displayed
- Negative
  only negative values will be displayed
- Positive/Negative
  both, positive and negative values will be displayed
- with offset
  the scale is shifted by an offset.

Offset:
specify an offset as an integer or real number in the field

Range:
The range of the vertical scale will be specified in powers of ten. If the scale shall extend for example from -100 to +100, type “2” in the Range field and select positive/negative in the Values box.

Style

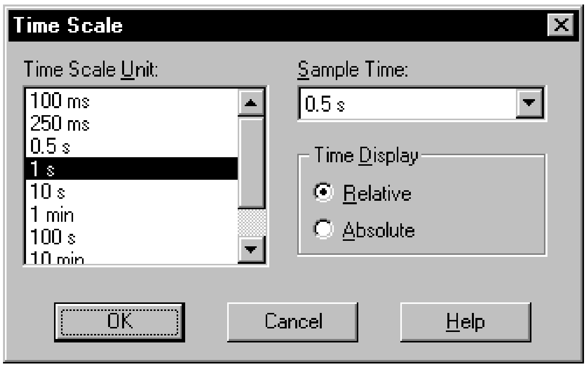
Line Style:
Three different styles for painting the thickness of the curve are available:

- Normal
- Bold
- Double Bold.

Color:
When pressing the **Color** button a color table will be displayed, from which you can select a color. The curve and the corresponding texts will be drawn with the selected color.

Time Scale

When pressing the **Time Scale** button the following dialog will be displayed:



Time Scale

Time Scale Unit:
100 ms
250 ms
0.5 s
1 s
10 s
1 min
100 s
10 min

Sample Time: 0.5 s

Time Display
● Relative
○ Absolute

Figure 3-4. Time Scale dialog

Dialog Controls of Time Scale

Time Scale Unit
Select one of the listed time units to be used for scaling the horizontal axis. The available times range from 100 ms to 1h.

Sample Time
The **Sample Time** specifies the cycle time for reading values from the Advant Controller 400 Series. A sample time of 10s means that the values will be read once in 10s.

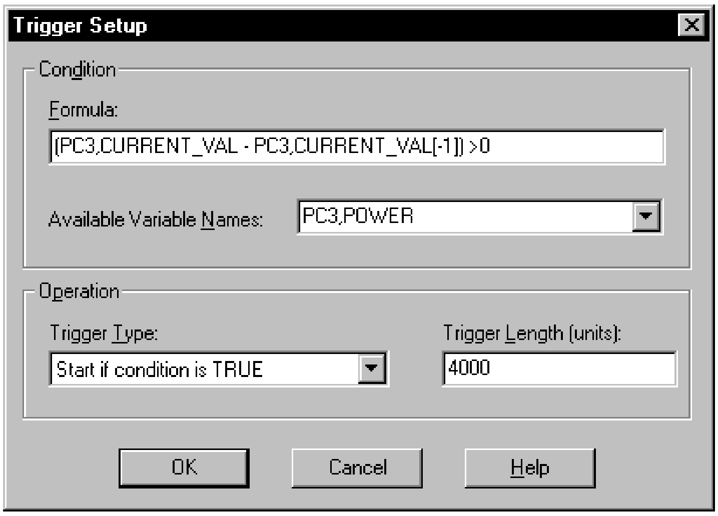
The **Sample Time** cannot be greater than the **Time Scale Unit**. Further on the number of display points is restricted. Therefore for long Time Scale Units the shorter Sample Times are not available.

If you select short **Sample Times** a warning message will be displayed. It informs you that the recording of data may lead to problems if the overall load on the Advant Controller 400 Series is higher than 90%.

Time Display
The **Time Display** can be either relative or absolute. Relative means that the time notation at the ticks of the time scale will be relative to the start of data recording. Absolute Time Display means notation according to the day time. The clock of the Advant Controller 400 Series will be used for the display.

Trigger

When pressing the **Trigger** button of the Setup dialog the **Trigger Setup** dialog will be displayed (see Figure 3-5):



Formula: (PC3,CURRENT_VAL - PC3,CURRENT_VAL[-1]) >0

Available Variable Names: PC3,POWER

Trigger Type: Start if condition is TRUE

Trigger Length (units): 4000

Figure 3-5. Trigger Setup dialog

With the **Trigger Setup** you can define an event and specify an action that shall take place when the event occurs. An example of an event may be that a signal exceeds a defined threshold. When this happens recording of values may start or stop. The detection of the event does not take place on the Advant Controller 400 Series, but on the engineering station with the data read from the controller.

Dialog Controls of Trigger Setup

Formula

The **Formula** is a valid expression which may contain values, constants, operators and an indexes for values.

Value: any value that was activated with the IV subcommand. You can pick up a name from the drop down box named Available Variable Names and insert it into the Formula field (via Copy/Paste).

Constant: any constant like 2 or 50.5

Operator: `<, <=, >, >=, +, -, * ,/, (, ), AND, OR, XOR, NOT.`

Index: value_name[n]; with $-10 \le n \le -1$; a value name without an index means the current value, Value_name [-1] means the foregoing value an so on.

Example: a formula to detect a positive slope of a curve may look like this:

```
(PC3,CURRENT_VAL - PC3,CURRENT_VAL[-1])>0
```

Available Variable Names

A list containing any value the was activated with the **IV** (Insert Value) subcommand. Use Copy/Paste to pick up values from this list and to insert them into the **Formula**.

Trigger Type

Possible values are:

- Disabled
  The trigger is disabled independent of the formula.
- Start if condition is TRUE
  means start recording when condition is true. Depending on the **Trigger Length** value, GETAB records the specified number of values after the event occurred. Then **GETAB** stops.
- Stop if condition is TRUE
  means stop recording when condition is true. Depending on the **Trigger Length** value, **GETAB** remembers at maximum the specified number of values before the event occurs. If the event occurs earlier, the number of recorded data points is according to the time between start of recording and the event.
- Center trigger event
  GETAB tries to keep the event centered in the amount of recorded data points. When for example the Trigger Length is set to 1000, **GETAB** tries to hold 500 data points before and 500 data points after the event. If the event occurs earlier, say at data point 200, **GETAB** will still record 800 data points after the occurrence of the event.

Trigger Length

The **Trigger Length** defines a continues area of data points around the triggered event. The length is specified in **Time Scale Units**. How the area is arranged around the triggered event depends on the **Trigger Type**.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.93 GFOR - Get FORmat

Syntax: **GFOR**

`GFOR mlevref`

This command is available at system level only.

Description: This command lists all properties of a specified format. Reference properties are just indicated in the listing, the properties they are referring to are not listed. Type HELP MLEVREF for the description of mlevref parameter.

3.94 GID - Get IDentity

Syntax: **GID**

This command is available at system level only.

Description: This command lists the data base instance that is currently under inspection. The listing includes all levels down to (including) the current level.

Example: *FINS 298,1;3=11

```
*GID
Concept    ! LF,LR ! Pathproperty   !  Mem ! Key
           !       !                !      !
           ! 298,1 !                ! 1    ! 3 = 11
```

Concept	Concept name (if available)
LF,LR	Logical File, Logical Record (set by the last FINS command)
Pathproperty	Accesspath to the current data structure within LF,LR
Mem	Member number for group properties
key	Search key (set by last FINS command).

3.95 GINS - Get INStance

Syntax: **GINS**

`GINS mlevref`

This command is available at system level only.

Description: This command lists all property values of the base level of an instance Reference properties are just indicated in the listing, the properties they are referring to are not listed. Type HELP MLEVREF for the description of mlevref parameter.

3.96 GLID - Get Last Item Designation

Syntax: **GLID**
This command is available at PC level only.

Description: This command displays the item designation in a PC program last referenced by the user during a session. It can be useful in establishing the item designation if the user uses only a partial designation in a command. See *Master Piece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: Get the last item designation referenced:

```
#GLID
PC 1.1.2
```

3.97 GOTO

Syntax: **GOTO** *label*

This command is available at PC and system level.

Description: This command passes execution control to a labeled line in a command file. “Label” is 1 to 10 characters terminated with a colon. See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.98 GPN - Get Page Number

Syntax: **GPN** *item_des*

This command is available at PC level only.

Description: This command displays the page number, and the first element in the page, on which the specified item designation is located. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: Get page where element PC2.1 is located:

```
#GPN PC2.1
```

```
PAGE NUMBER: 9

FIRST ITEM DES: PC2
```

3.99 GPRO - Get PROperty

Syntax: **GPRO** *mlevref*

This command is available at system level only.

Description: This command lists the value of the last property in the multi level reference. Type HELP MLEVREF for the description of mlevref parameter.

3.100 HARDCOPY

Syntax: **HARDCOPY** [**ON**|**OFF**]

This command is available at PC and system level.

Description: This command can be used to route data from the engineering station screen to the printer connected to it. It can be used in three ways:

- Copy current contents of screen to printer (HARDCOPY)
- From now on copy all incoming data to printer (HARDCOPY ON)
- Stop copying data to printer (HARDCOPY OFF).

This command is not available with the Engineering Board PU516 and the RTA board.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.101 HELP

Syntax: **HELP** [**HELP**|/**COMMANDS**|/**CMD**|/**ALL**]

`HELP` `[`*`command_name`*`|`*`keyword`*`]`

This command is available at PC and system level.

Description: The HELP function presents information concerning commands available, their syntax and how to use them. Other information about system characteristics and details for system users are returned by using relevant keywords.

Examples with qualifiers:

To get general information about HELP, type:

```
#HELP
```

or

```
#HELP HELP
```

To get a list of available commands, type:

```
#HELP /COMMANDS
```

or

```
#HELP /CMD
```

To get a list of available commands and keywords, type:

```
#HELP /ALL
```

Examples with parameters:

To get the help text of a command, for example **LDBD**, type:

```
#HELP LDBD
```

To get the help text of a keyword, for example LEXICALS, type:

```
#HELP LEXICALS
```

You can get a list of available PC elements or DB elements (when connected with an Advant Controller 400 Series):

```
#HELP DBEL
#HELP PCEL
```

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.102 IF

Syntax: **IF** *symbol|expression comp-operator expression* **THEN** *command*

This command is available at PC and system level.

Description: This command executes “command” if the given statement is true. “comp-operator” is one of (.EQ. .GE. .GT. .LE. .LT. .NE. .EQS. .NES.). See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.103 INCPT - INCrease Program Tables

Syntax: **INCPT** *pc_propgram*
This command is available at PC level only.

Description: This command is redundant for certain system types. For those systems, the command terminates after a printout about this. The command is used to display and increase the empty size of a PC program and its five data tables. The empty area in the PC program may be increased by moving spare memory from the system empty area. The five PC program tables can be increased by moving some of the memory in the PC program empty area. This operation can only take place if the PC program is in build mode.

Example:

Call command with pc_pgm PC8:

```
#INCPT PC8
```

3.104 INQUIRE

Syntax: **INQUIRE** *text,symbol*[;**NOCONV**]
This command is available at PC and system level.

Description: This command is used to get information from the user of a command file. “Text” (enclosed with double quotes) is printed and the answer is assigned to “symbol” (local). The characters in the answer are converted to uppercase, if not “NOCONV” is given. See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.105 INSTASK - INStall TASK

Syntax: **INSTASK** [*volume*:*segment*][;**L**]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command adds new or modifies existing tasks in a target system. volume and segment specifies a segment on a disc containing the new task code. If option “L” is specified, messages are written on the console after each phase the command executes. Type HELP INSTALL for more information.

This command is not available with the Engineering Board PU516 and the RTA board.

3.106 INTERACTION

Syntax: **INTERACTION**

This command is available at PC and system level.

Description: This command directs input from a command file to the terminal during the next user command. See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.107 IS - Insert Statement

Syntax: **IS** *item_des*[;**END**]

This command is available at PC level only.

Description: This command enables:

- Construction of a new program.
- Entry of one or more statements into an existing program after the specified item_designation (item_des).
- Entry of one or more statements into an existing program last in the specified module by using ;END in the command line.

The following subcommands are available:

<Enter>	Moves to the next item designation (within the proposed level)
<<	Moves up to a higher program level.
!	Ends the dialog
>S	Assigns a name to an instance
P	Add soft page break
Px	Add hard page break; x is page number to be used.
P/H	Add soft page break and headline

Px/H	Add hard page break and headline

A hard page break introduces a new page with the page number specified. A page created with a soft page break gets the next consecutive number. See *MasterPiece 200/1 Configuration and Programming* or *AMPL Configuration Advant Controller 400 Series Reference Manual*.

Example 1: Entry of additional statements:

```
#IS PC1.1.6
PC1.1.7? <<
PC1.2? CONTRM(50)
PC1.2.1? FUNCM
PC1.2.1.1? SR-00(2,2)!
```

Example 2: Insert at the end of PC program:

```
#IS PC1;END
```

Example 3: Insert at the end of control module PC1.2:

```
#IS PC1.2;END
```

Example 4: Insert an additional statement and assign a name for the instance:

```
#IS PC1.1.6
PC1.1.7? AND(2)>S
INAME=BinaryLogic
PC1.1.8?
```

3.108 KERMIT

Syntax: **KERMIT** [*Kermit_subcommand*]
This command is available at PC and system level.

Description: The command KERMIT starts a file transfer program which uses the popular Kermit file transfer protocol. The parameter Kermit_subcommand is any subcommand which may be entered at the Kermit subcommand prompt. This parameter is optional. When the command KERMIT is entered, the Kermit program will first execute the Kermit subcommand given as parameter and then prompt for more subcommands. To exit from Kermit give the subcommand EXIT. To get information about the available Kermit subcommands give the subcommand HELP at the Kermit subcommand prompt.

This command is not available with the Engineering Board PU516 and the RTA board.

3.109 KIT - KIll Task

Syntax: **KIT** *task_name*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command kills a task. Type HELP TASK_NAME for description of task names.

Example: To kill the task DEFS003:
KIT DEFS003

3.110 LAABF - Load AAb File

Syntax: **LAABF** [*volume*]:*segment,name*/*option*
This command is available at PC and system level.

volume	Source volume; default is SRCE
segment	Segment to be loaded
name	Name of the data base element to which the data should be loaded. Element_name can be either the element name or the element id
option	UNLOCK - unlock a file that is still locked

Description: This command is used to copy an AAB file to the data base. If the command is given without parameters, parameters are requested in a dialog.

Example 1: Load file AABONMA2.DD to data base element AAB_FILE1:

```
#LAABF SRCE:AABONMA2.DD,AAB_FILE1  or
#LAABF AABONMA2.DD,AAB_FILE1
```

Example 2: Load file AABONMA2.DD to data base element with element name MY_AAB_FILE.

```
#LAABF SRCE:AABONMA2.DD,MY_AAB_FILE  or
#LAABF AABONMA2.DD,MY_AAB_FILE
```

3.111 LAVOL - List AVailable VOLumes

Syntax: **LAVOL**
This command is available at PC and system level.

Description: List all volumes that can be accessed by the engineering station within the session.

3.112 LC - List Channels

Syntax: **LC**
This command is available at system level only.

Description: This command lists the channels available in the system.

3.113 LCI - List Channel Information

Syntax: **LCI** [*channel_name*]
This command is available at system level only.

Description: This command lists information related to a channel. “channel_name” is name of channel (1 to 6 characters). If the channel name is omitted, the last name entered is assumed.

Example 1: To list information related to channel SYSCHA:
LCI SYSCHA

Example 2: To list information related to last entered channel:
LCI

3.114 LCT - List Concept Table

Syntax: **LCT**
This command is available at PC and system level.

Description: This command is useful to programmers who have added their own data base elements and wish to inspect the concept table values.

3.115 LCU - List Channel Utilization

Syntax: **LCU** [*utilization_degree*]
This command is available at PC and system level.

Description: Lists the channel utilization for the channels in the system. For each channel the “SIZE” (channel size) and “USED” (number of queued signals) are given. The command has two forms:

1. List channel utilization for all channels in the system.
2. List channel utilization for those channels with an utilization degree higher as the specified utilization degree. The utilization degree is specified as a per value between 0 - 100, no decimals are allowed.

Example:To list the channel utilization for those channels that are half filled or more than half filled: **LCU 50**
This command is not available with the RTA board.

3.116 LD - List Debug points

Syntax: **LD** [*debug_no*][,*debug_no*]..[,*debug_no*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.
This command is available at system level only.

Description: This command lists information about debug points. “debug_no” is debug point on which information is needed. If none specified all the defined debug points are assumed.

Example: To list information about debug points 2 and 6:
LD 2,6

3.117 LDBD - List Data Base Diagram

Syntax: **LDBD**

LDBD SEL=*select*[=*start*][,*interv*][,*gap*][/**XREF**[;*PCm*[>*PCn*]]]

LDBD SEL=*select*[/**X_ONLY**[;*PCm*[>*PCn*]]][/**S_ONLY**]

This command is available at PC and system level.

Description:

interv:	Page interval between elements of the same type.
gap:	Page interval between elements of a different type.
/XREF:	Cross references between PC programs and data base will be added to the listing. If no PC program is specified, all PC programs will be scanned.
/X_ONLY:	Only terminals containing cross references to PC programs will be listed. This switch can be combined with **S_ONLY**.
/S_ONLY:	Only the summary will be listed. This switch can be combined with **X_ONLY**.

Normally the output medium is the printer. But if **X_ONLY** or **S_ONLY** is specified, the output medium depends on the **SEOM** settings and no header and footer text is printed.

Examples on the “select” parameter:

- Selects all data base elements in every category, (SEL=ALL).
- Selection of data base elements using wildcards, (e.g. SEL=A*, which selects all data base elements starting with “A”); the wildcard “*” allowed at the end of the string only.
- Selects all data base elements of one type, (e.g. SEL=DIC).
- Selects a range of elements of the same type, (e.g. SEL=AI1>4).
- Selects one specific data base element. (e.g. SEL=AIC6).

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: List all DI Calculated elements; First page = 10:
#LDBD SEL=DIC=10

Example 2: List all data base elements; First page = 100: Page interval between elements of the same type = 10: Page interval between different element types = 50:
#LDBD SEL=ALL=100,10,50

Example 3: List the first 3 elements of type DI Calculated:
#LDBD SEL=DIC1>3

Example 4: List all data base elements and all cross-references from PC1 to PC3
#LDBD SEL=ALL/XREF;PC1>PC3

Example 5: Use of wildcards within the selection:
#LDBD SEL=A*
This command lists all data base element instances starting with “A”, e.g. AI, AO, AIPT, AIC

NOTES

- LDBD is an interactive command and will prompt the user for all required input, including the initial parameters (if omitted).
- The first element in the range MUST exist. In Example 3, DIC1 must exist.
- A single subordinate element cannot be listed. If you wish to list AI3.4, the elements AI3 and all its related subordinate elements AI3.1 - AI3.16 must also be listed.

3.118 LDEV - List DEVices

Syntax: **LDEV**

This command is available at system level only.

Description: This command lists the devices available in the system.

3.119 LDEVI - List DEVice Information

Syntax: **LDEVI** [*device_name*]

This command is available at system level only.

Description: This command lists information related to a device. “device_name” is name of device (1 to 6 characters). If the device name is omitted, the last name entered is assumed.

Example 1: To list information related to device PRINTE:

```
LDEVI PRINTE
```

Example 2: To list information related to last entered device:

```
LDEVI
```

3.120 LDIR - List DIRectory

Syntax: **LDIR** [*volume*]:[*segment*]

This command is available at PC and system level.

Description: This command lists directory information for given segments. “segment” is segment(s). If only character “:” is specified information for all segments is listed. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Example: To list info on segments on volume XAXS with extension RO:

```
LDIR XAXS:*.RO
```

3.121 LDISP - List DISPlay information

Syntax: **LDISP** *P1*[,*P2*,*P3*,*P4*,*P5*][>*P6*][/*P7*,*P8*][:*P9*]
This command is available at system level only.

Description: LDISP lists information from concept DISPLAY_DES, DYNAMIC_PART_DES and SYMBOL_TABLE. See documentation of *Advant Station 500 Series Operator Station User’s Guide* or *MasterView 800/1* for more information. Parameters:

P1-P5:	Display name.
P6:	Display name. Last display in interval.
P7:	First dynamic point to be listed. Default value is 1.
P8:	Last dynamic point to be listed. Default value is 255.
P9:	Options (see below).
B (brief):	Lists for example static and dynamic parts, function keys and xy dialogs.
L (long):	Lists for example. static and dynamic parts, dialogs, dynamic points, logical file and logical instance.
P (printer):	Data will be listed on printer connected to the engineering station.

Example 1: **LDISP PROC.01**
Lists contents of the display PROC.01

Example 2: **LDISP PROC.02,OVW.01**
Lists contents of the displays PROC.02 and OVW.01

Example 3: **LDISP PROC.03>PROC.07/1,10**
Lists contents of the display PROC.03 through to PROC.07 about dynamic point 1 through to point 10.

Example 4: **LDISP GRP.1.03:B**
Lists contents of the display GRP.1.03 in brief mode.

Example 5: **LDISP PROC.*:LP**
Lists contents of all PROC.XX displays starting with PROC.01, in long, print modes.

3.122 LDURWM - List info about RWM dump

Syntax: **LDURWM** *volume*:*segment*

This command is part of the product Expert Utilities and only available at system level.

Description: Llists information about a backup done with command DURWM. A dump created by command DURWM has a segment holding information about the dump, which can be listed using command LDURWM. This is included: dump description, dump size, name of first and next volume included. Type HELP VOLUME_NAME for description of volume name. Segment is 1-7 characters, the eight being used for numbering the segments. Note that extension should not be given. Example:

```
LDURWM VOLA:BACKUPA
```

Current volume is VOLA. Segment BACKUPAI.XI holds information about the dump.
This command is not available with the Engineering Board PU516 and the RTA board.

3.123 LGAM - List Group Alarm Members

Syntax: **LGAM**

```
LGAM [SEL=select[=[start_page_numb][,page_interv]][/EXP]]
```

This command is available at PC and system level.

Description: Start_page_numb Page number for the first element.

Page_interv page interval between elements.

/EXP expanded list of Group Member.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Examples on the “select” parameter:

- Selects all data base elements in Group Alarm category, (SEL=ALL).
- Selects a range of Group Alarm elements, (e.g. SEL=GRPALARM1>4).
- Selects one specific data base element, (e.g. SEL=GRPALARM6).

Example 1: List all GRPALARM elements; First page = 10:

```
#LGAM SEL=ALL=10
```

Example 2: List Group Alarm elements 1-5; First page = 100; Page interval = 10:

```
#LGAM SEL=GRPALARM1>5=100,10
```

Example 3: List all GRPALARM elements; First page = 10; Expanded list = /EXP:

```
#LGAM SEL=ALL=10/EXP
```

NOTE

LGAM is an interactive command and will prompt for all required input, including the initial parameters, if omitted.

3.124 LIF - List Inhibit Flags

Syntax: **LIF**

This command is available at PC level only.

This command lists all PC programs with inhibit flag set. The flag can be set with the SEIF command.

See also: **RSIF**, **SEIF**, **SWBL**.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.125 LIQB - List I/O Queue Blocks

Syntax: **LIQB** *task_name*

This command is available at PC and system level.

Description: This command lists the IQBs (I/O Queue Blocks) for selected task.

This command is not available with the RTA board.

3.126 LIST - LIST segment

Syntax: **LIST** [*volume*]:[*segment*][=*line1*][>*line2*][;*option*]

This command is available at PC and system level.

Description: This command lists a segment. “segment” may contain wild cards.

Line1:First output line number. Default is 1.

Line2: Last output line number. Default is last line of segment.

Option C: Console is output medium.

Option P: Printer is output medium. Default output medium is C.

Option #: Line numbers are included in the output.

Option N: No header.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Examples: To list the file “HELP.SA” from volume “HLP1”:

```
LIST HLP1:HELP.SA
```

To output lines 3 to 5 of all files of the type HLP1:*.%A on both printer and console, with line numbers and without header:

```
LIST HLP1:*.%A=3>5;P,C,N,#
```

3.127 LLUA - List Logical Unit Assignments

Syntax: **LLUA**

This command is available at system level only.

Description: This command lists the currently logical unit assignments in the system.

3.128 LMFS - List MasterFieldbus Status

Syntax: **LMFS**

This command is available at PC and system level.

Description: This command shows the communication status for busses on MasterFieldbus. To check the communication quality the user has to study the error counter values and compare the difference between two readings. If a fatal error is detected on some node the error counter value is not displayed. Target system must be in operational mode and SIO_INTERFACE must exist in the data base.

3.129 LMP - List Modify Permission

Syntax: **LMP** *item_des*[>*item_des*]

This command is available at PC level only.

Description: This command displays if the modify permission is enabled or disabled in one or more execution units. The two item designations must refer to a PC program or an execution unit (PCPGM, CONTRM, MASTER, SEQ).

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: Call command with execution units PC2.2, PC2.3, PC2.4:

```
#LMP PC2.2>PC2.4
```

3.130 LOADFILE

Syntax: **LOADFILE** *name*[,*filename*][/FUZZYCON]

where

nameName of the data base element of type **AAB_FILE** into which filename shall be loaded.

filenameEither: volume:segment.extension

Or: a valid DOS path and filename

/FUZZYCON tells the command that the file contains a configuration for the fuzzy control PC elements.

This command is available at PC and system level.

The **LOADFILE** command loads the contents of *filename* into the data base element *name* of type **AAB_FILE**. It is needed when **User Defined PC Elements** shall be loaded to Advant Controller 400 Series. For more information see *User Defined PC Elements User's Guide.*

3.131 LOAP - LOad APplication

Syntax: **LOAP** *dump_name*[,*volume*][;*options*]

where

dump_name	1 to 7 alphanumeric characters
options	a string of one or more option identifiers
volume	SRCE or USER

This command is available at PC and system level.

The dump is loaded from the volume SRCE if `volume` is not specified. The dump description appears on the screen and you decide whether to continue with loading. Option identifiers:

I:	Gives more progress texts than by default.
U:	Allows loading even if network/node numbers differ in dump and target. By default the network/node numbers must agree.

This command is available at PC and system level.

Description: This command is used to load a DUAP backup dump into an Advant Controller 400 Series. The DUAP dump contains all data base and PC applications from the dumped target.
The command works in remote session and configuration mode, and in off-line session. Remote session can run on levels System and PC.

See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: **LOAP MYDUMP;I**
Load command in dialog mode with increased progress texts. The operator will be queried for volume names etc.

Example 2: **LOAP APDUMP,SRCE**
Load command in parameter mode. The parameter mode is primarily intended to be used when the command is run by a command file. It requires no operator interaction. If an undefined situation occurs the command terminates with an error message.

3.132 LOCK

Syntax: **LOCK** *pc_pgm*
This command is available at PC level only.

Description: This command can be used to set a password on a PC program to avoid unauthorized listing, reading and changing of the PC program. One or more PC programs can be locked without disturbing the execution. A password must be at least four but no longer than ten characters. Digits, upper and lower case letters and underscore are allowed in a password. Upper and lower case letters are not considered equal.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: **#LOCK PC1**
Password: _________
Verification: _________

3.133 LOCON - LOad CONcept(s)

Syntax: **LOCON**
LOCON *volume*:*segment*(;*options*)

WARNING

This command shall be used by experts only!

This command is available at PC and system level.

Description: volume: Volume name, 4 characters

segment: Segment name, max. 8 characters, no extension, may contain wildcards

options:
C = load data base in compressed mode
5 = expand INDSV-file with an extra 5%
F = force loading; ignore all warnings
I = display more information

If this command is used without parameters, it starts a dialog, in which the user can specify the name of the dump volume and then start loading the data base dump into the target system.

If volume and segment name are specified in the command line, all files with the extension .CD which match the segment specification are loaded.

Example 1: Load the dump segment KLI01.CD from the volume SRCE.
***LOCON SRCE:KLI01**

The segment KLI01.CD is loaded. The data base will not be compressed.

Example 2: Load the segments TABLE01.CD ... TABLEnn.CD from volume SRCE.
***LOCON SRCE:TABLE%%**

The segment(s) TABLE01.CD ... TABLEnn.CD are loaded.

Example 3: Load the segments TABLE01.CD ... TABLEnn.CD from volume SRCE. Compress the data base while loading.
***LOCON SRCE:TABLE%%;C**

The segment(s) TABLE01.CD ... TABLEnn.CD are loaded. The data base is compressed.

Example 4: Load the segments starting with the name ALL from the volume SRCE. Compress the data base while loading. Expand INDSV-file with an extra 5%. Ignore all warnings, but display extended load information.
***LOCON SRCE:ALL*;C5FI**

3.134 LOCPSET - LOad CP SET

Syntax: **LOCPSET** *CPfile*[;**V**]

Description: This command loads and initiates a set of Command Processors. Any previously loaded set is removed. “CPfile” is a text file (volume:segment.extension) containing file specifications (volume:segment.extension) of Command Processors to be loaded (one specification/line). Option “V” (verbose) displays progress information.

This command is not available with the Engineering Board DSPU131.

3.135 LODB - LOad Data Base

Syntax: **LODB** *volume*:*segment*
LODB *volume*:*segment*;*****
LODB *volume*:*segment*;*logic_file*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command can be used to load a Logical dump. If the third parameter is not given, you will be asked to enter it. Type HELP VOLUME_NAME for description of volume. Type HELP SEGMENT_NAME for description of segment. See documentation of *Advant Station 500 Series Operator Station User’s Guide* or *MasterView 800/1* for more information.

3.136 LORWM - LOad RWM

Syntax: **LORWM** *volume*:*segment*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.
This command is available at system level only.

Description: This command loads a system’s RWM with data previous dumped using command DURWM. The volume and segment given in the command must hold the first part of the dump.
The loading can only be done in an off-line session, i.e. the system must have its rudimentary monitor running. The system’s RWM is cleared before the loading starts. Type HELP VOLUME_NAME for description of volume name. Segment is 1-7 characters, the eight being used for numbering the segments. Note that extension should not be given. The command can be aborted by pressing **<Ctrl-Break>**.

Example:**LORWM VOLA:BACKUPA**
The loading starts with volume VOLA. Segment BACKUPAI.XI holds information about the dump and BACKUPA1.XD the first part of data.
This command is not available with the Engineering Board PU516 and the RTA board.

3.137 LOTDB - LOad Total Data Base

Syntax: **LOTDB** [*dump_name*]
dump_name1...6 characters
This command is available at PC and system level.

Description: The command is used to load a total dump of a data base into the target. The dialog is self instructive and the valid answers are listed within the prompting text of every question.
The mass memory volumes holding the dump are first checked by the command. In case of approval, the operator is queried weather the dump shall be loaded in compressed or in normal mode and whether the target data base shall be redimensioned or not. For the redimension dialog, please refer to the DIMDB command for instructions. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* more information.

3.138 LOTPT - LOad Total Program Table

Syntax: **LOTPT** [*dump_name*]
LOTPT [*dump_name*, *volume*]
This command is available at PC level only.

Description: dump_name name of the dump, 6 characters
volume name, 4 characters

This command is used to load an earlier dumped system. The command operates in dialog mode. Enter **<Ctrl-Break>** to break.

NOTE

A PC dump from an Advant Controller 400 Series system / MasterPiece 200/1 system can only be loaded in a similar system.

The command will load the dump from the volume SRCE by default.
See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.139 LPCD - List PC Diagram

Syntax:

LPCD [*pc_pgm*,] *page_nr* [>*page_nr*]
LPCD [*pc_pgm*,] *page_nr* [;*number_of_pages*]
LPCD *pc_pgm* [>*pc_pgm*]
LPCD *pc_pgm* [;*number_of_pc_pgms*]
This command is available at PC level only.

Description: This command is used to list the PC diagram. Before running LPCD use the commands:

- SEOM to set output medium to printer and/or console.
- SEPS to set the page size for the PC diagram.
- MPD, MPN to define page divisions.
- SLNUM or SLSYM to select numeric or symbolic references.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.140 LPCL - List PC List

Syntax:

LPCL *pc_pgm*[>*pc_pgm*][=*page_nr*]
This command is available at PC level only.

Description: This command is used to print a functional list of one or more PC programs. The format of the PC list should first be defined using the SEPS (SEt Page Size) command.
See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: List PC list for PC programs PC1, PC2, PC3 and PC4:
#LPCL PC1>PC4

Example 2: List PC list for PC program PC6 starting with page 19:
#LPCL PC6=19

3.141 LPI - List Port Information

Syntax: **LPI** [*task_name*]
This command is available at system level only.

Description: Lists information related to the inports and the outports of a task. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.

Example 1: To list information related to ports of task DEFS003:
LPI DEFS003

Example 2: To list information related to ports of last entered task:
LPI

3.142 LPN - List Page Number

Syntax: **LPN** [*pc_pgm*,][*page_nr*[>*page_nr*]]
`LPN [pc_pgm,][page_nr[;number_of_pages]]`

This command is available at PC level only.

Description: This command displays the total number of pages in the PC program, and the page number and first element on the page, for pages specified in the command.
Soft page breaks are marked with ’#’.
See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: List the pages containing item designation PC1.1.2:

```
#LPN PC1.1.2,
NUMBER OF PAGES IN TABLE : 6
PAGE NUMBER : 2
FIRST ITEM DES : PC1.2
```

Example 2: List 2 pages starting with page 3:

```
#LPN 3;2
NUMBER OF PAGES IN TABLE : 6
PAGE NUMBER : 3
FIRST ITEM DES : PC1.3
PAGE NUMBER : 4
```

3.143 LS - List Statement

Syntax: **LS** *item_des*[>*item_des*]
LS *item_des*[;*number_of_elements*]
This command is available at PC level only.

Description: This command is used to list:

- A single statement.
- All statements between and including the two item designations.
- A number of statements beginning with the item designation specified.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: List statements PC1.1 to PC1.1.6:

```
#LS PC1.1>PC1.1.6
```

Example 2: List 4 statements starting with PC1.1.1:

```
#LS PC1.1.1;4
```

3.144 LSIG - List SIGnals

Syntax: **LSIG** [*channel_name*[;**ALL**]]

```
LSIG [channel_name[;signal_number]]
```

Description: This command lists the signals or information about the signals that are queued in one or in all channels. If the second parameter is given then the actual signal(s) are listed in a hexadecimal format. The command has the four forms:

- Give a short information about each signal inside all channels.
- Give a short information about each signal inside the selected channel.
- Give a short information about each signal inside the selected channel and display all these signals in a hexadecimal format.
- Give a short information about each signal inside the selected channel and display the selected signal in a hexadecimal format.

NOTE

When this command is used in a running system signals may pass through a channel while these signals are displayed. This will cause trouble. If the “LSIG” command notices that data in a channel is changing then the logging from this channel will immediately be stopped. However, it might happen that the command does not notice a change. A displayed signal can then be a mix of several signals.

Example 1: To list a short information about each signal inside the channel CXCCH2:

```
LSIG CXCCH2
```

Example 2: To list a short information about each signal inside channel CXCCH2 and to display the signal queued as number 2 in this channel:

```
LSIG CXCCH2;2
```

This command is not available with the RTA board.

3.145 LSYSHI - List SYStem Halt Information

Syntax: **LSYSHI**
This command is available at system level only.

Description: This command lists certain important information about the system. It is useful for analyzing a halted system.

This command is not available with the RTA board.

3.146 LSS - List Special Status

Syntax: **LSS**
LSS *item_des*[>*item_des*]
This command is available at PC level only.

Description: This command enables the user to list the special status for:

- A single execution unit (PCPGM, MASTER, CONTRM, SEQ).
- A group of execution units.
- All executions units in the system.

The special status types are:

- Unblocked (Complete PC program).
- Build mode (Complete PC program).
- Blocked.
- Blocked for data base.

The mode of the system (configuration or operation) is also listed.
See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: List special status of execution unit PC1.2:

```
#LSS PC1.2

System in configuration mode
BUILD MODE : PC1
```

Example 2: List special status of execution units from: PC1 to PC2.5 System in operation mode

```
#LSS PC1>PC2.5
BLOCKED: PC1.9
BLOCKED FOR DB: PC2.5
```

Example 3: List special status of all execution units in the system: System in operation mode

```
#LSS
BLOCKED FODB: PC1.1
BUILD MODE: PC2
UNBLOCKED: PC3
```

3.147 LSYSM - List SYStem Messages

Syntax: **LSYSM** [**ALL**]
This command is available at PC and system level.

Description: This command lists remaining system messages that still are in channel SYSCHA. It is valuable to use when analyzing a halted system. The command has two forms:

1. List all remaining system messages that are queued in the system message channel.

As above and if possible list already printed system messages.
This command is not available with the RTA board.

3.148 LT - List Tasks

Syntax: **LT**

This command is available at system level only.

Description: This command lists the tasks currently present in the system.

3.149 LTI - List Task Information

Syntax: **LTI** [*task_name*]
LTI [*tcb_addr*]
This command is available at system level only.

Description: This command lists selected information from a task control block (TCB) and program description block. If the task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names. In the alternate syntax for the command, “tcb_addr” is the start address for the task control block. All addresses stated in the listing are absolute addresses. Sizes are given in bytes.

Examples Example 1:

To list information related to task DEFS003:

```
LTI DEFS003
```

Example 2: To list information related to last entered task:

```
LTI
```

3.150 LTIMQ - List TIMer Queues

Syntax: **LTIMQ** [*queue*]

This command is available at PC and system level.

Description: This command list the contents of one or all timer queues. The parameter “queue” is one of “TQ”, “DQ” and “DTQ”. The command has the four forms:

1. List all timer queues.
2. List the “Timer Queue”.
3. List the “Delay Queue”.
4. List the “Diagnostic Timer Queue”.

Example: To list the contents (queued tasks and signals) of the “Delay Queue”:

```
LTIMQ DQ
```

This command is not available with the RTA board.

3.151 LTREE - List TREE

Syntax: **LTREE** *item_des*[>*item_des*][/*mode*]

LTREE *item_des*[;*number_of_elements*][/*mode*]

This command is available at PC level only.

Description: This command lists a tree structure of the specified part(s) of PC program(s).

Soft page breaks are marked with ’#’ if mode is L(long) or S(short).

Mode parameter values:

Normal mode: All PC elements.

S	Short mode: Structural PC elements + PC diagram headlines and page no.
L	Long mode: All PC elements + PC diagram headlines and page no.
D	Normal disk mode: Normal mode + Diskette file.
SD	Short disk mode: Short mode + Diskette file.
LD	Long disk mode: Long mode + Diskette file.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Examples: List TREE of PC1.1.5 and subelements:

```
#LTREE PC1.1.5;
```

List TREE of the elements PC1.1.5 - PC1.1.10. Add PC diagram headlines and page no:

```
#LTREE PC1.1.5>PC1.1.10/L
```

List TREE of PC1 and subsidiary structural elements. Add PC diagram headlines and page no. Destinate the output also to diskette:

```
#LTREE PC1/SD
```

3.152 LTRQ - List Task Ready Queues

Syntax: **LTRQ** [*priority*][,*priority*]..[,*priority*]

This command is available at PC and system level.

Description: This command lists the tasks in the ready queues. It can only be used to analyse a halted (off-line) system and it has two forms:

1. List all tasks in ready queue 0 - 15.
2. List all tasks in one or several ready queues.

Example: To list the tasks in ready queue 3 and 5:

```
LRQ 5,3
```

This command is not available with the RTA board.

3.153 LTSL - List Task Switch Log

Syntax: **LTSL** [*HEX*]

This command is available at PC and system level.

Description: This command lists the contents of the task switch log buffer. It has two forms:

1. List all tasks inside the task switch log buffer in a symbolic form.
2. List the contents of the task switch log buffer in hexadecimal format.

This command is not available with the RTA board.

3.154 LV - List Variable

Syntax: **LV** *item_des*:*terminal*

LV *[pc_pgm,]variable_name*

LV *[pc_pgm,][***DB***]=database_instance[:terminal]*

This command is available at PC level only.

Description: This command displays the element terminals connected to the specified point. This can be a terminal, a PC variable or a point in the data base. If a variable name but no PC program number was given the command assumes last PC program used. If data base instance and PC program were given this command displays all connections to the specified PC program and a list of other PC programs which have connections to the data base instance. If only data base instance was given this command displays all connections in all PC programs to this instance.

NOTE

If there is no terminal given with the *database_instance* the **VALUE** terminal is used as default connection point. If you want a listing of all *database_instance* terminals including PC connections, use **LDBD SEL=***database_instance*/**XREF**. You can reduce the size of the listing by using

LDBD SEL=*database_instance***/X_ONLY**, which excludes terminals without PC connections.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Call the command with a terminal identity:

```
#LV PC1.1.4:20
```

Example 2: Call the command with a variable name:

```
#LV PC1,ELEVATOR_UP
```

Example 3: Call the command with a data base element and the data base element’s terminal:

```
#LV PC1,DB=DO1.8:VALUE
```

Call the command with a data base element, the terminal VALUE is used as default:

```
#LV DB=DO1.8
```

3.155 MDB - Modify Data Base

Syntax: **MDB**

```
MDB element-name
MDB item-designation
MDB logic-file.logic-rec
MDB bus,station[,position[,subposition]]
```

This command is available at PC and system level.

Description: This command is used to inspect or modify the properties of a single data base element. MDB is an interactive command and will prompt the user for all required input, including the initial parameter (if omitted). See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.156 MDT - Modify Document Texts

Syntax: **MDT** *pc_pgm*[>*pc_pgm*][=*pc_pgm*]

```
MDT DB
```

This command is available at PC level only.

Description: With MDT you can read, copy or modify the document texts for either PC or data base listings. If the third parameter is specified, the associated document texts will be copied to the destination given with the first or second parameter. If the third parameter is omitted, you can enter the new document texts within a dialog.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Examples: Copy the texts from PC5 to all PC programs PC7, PC8, and PC9:

```
#MDT PC7>PC9=PC5
```

Modify the document text of PC program 7:
`#MDT PC7`

3.157 MFOR - Modify FORmat

Syntax: **MFOR**

`MFOR` *`mlevref`*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command modifies all properties for a specified format. The value of the specified property will be listed and the new value can be supplied. If **<Enter>** is pressed then the old value will be preserved. Type HELP MLEVREF for the description of mlevref parameter.

3.158 MINS - Modify INStance

Syntax: **MINS**
MINS *mlevref*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: Modifies all properties of a specified instance. Without a multi level reference, the record defined by the command FINS will be modified. The value of each property will be listed and the new value can be supplied. If **<Enter>** is pressed then the old value will be preserved. Press <**!**> to leave MINS directly. Type HELP MLEVREF for the description of mlevref parameter.

3.159 MNV - Modify Name of Variable

Syntax: **MNV** *item_des*:*terminal*
MNV [*pc_pgm*,]*variable_name*
MNV item_des
MNV [pc_pgm],[instance name]
This command is available at PC level only.

Description: This command is used

- to enter, read, change and delete variable names. Variable names can be defined for:
  - Element outputs.
  - Operational parameters and user defined constants.
  - Inputs coupled to the data base (not the structure element inputs.
- To enter, read, change and delete Instance Names which can be defined for PC elements.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Call the command with a terminal identity:
#MNV PC1.1.1:20

Example 2: Call the command with a variable name:
#MNV PC1,DRIVE_ON

Example 3: Read and delete variable name RELAY_ON in the same PC program:
#MNV RELAY_ON

```
PC1.1.1:2 /RELAY_ON= .
```

Example 4: Call the command with a PC item designation; an instance name for the specified item designation is added/modified
#MNV PC10.1

```
PC10.1
/
=ControlElement
```

Example 5: Call the command with an instance name for the specified item designation and delete the instance name

```
#MNV PC10,ControlElement
PC10.1
/ControlElement
=.
```

3.160 MPD - Modify Page Division

Syntax: **MPD** *item_des_1*[>*item_des_2*]=*page_nr*[/**H**]

`MPD item_des_1[;number_of_elements]=page_nr[/H]`

This command is available at PC level only.

Description: This command can be used to:

- DEFINE the start of a new page,
- MODIFY a page division,
- DELETE a page division.

for the specified PC program(s) in preparation for the LPCD command. It will convert existing soft page breaks into hard page breaks. One exception: If you modify only one item designation to which a soft page break is assigned, then this soft page break will not be converted.

PAGE DEFINITION

New pages are defined by specifying a single item designation. For example, if a PC program contains PC1, PC1.1 and PC1.1.1 through PC1.1.30, pages 1, 2 and 3 can be initially defined to start at PC1, PC1.1.10 and PC1.1.20 respectively:

Example 1: **#MPD PC1=1**

Example 2: **#MPD PC1.1.10=2**

Example 3: **#MPD PC1.1.20=3**

After initial definition all elements following the specified item designation are also allocated to the specified page. For example, after MPD PC1=1 the entire PC program will be allocated to Page 1, until MPD PC1.1.10=2 when PC1.1.10 onwards will be allocated to Page 2.

PAGE MODIFICATION

Page divisions are modified by allocating a different range of item designations to a page. However, as every item designation must at all times be allocated to a page, modifications must be performed in a set order. Following on from EXAMPLE 3, the divisions for pages 2 and 3 can be modified to start at PC1.1.5 and PC1.1.15 respectively:

Example 4: **#MPD PC1.1.15>PC1.1.30=3**

Example 5: **#MPD PC1.1.5>PC1.1.14=2**

Page 3 is modified before Page 2 to prevent item designations PC1.1.15 through PC1.1.19 becoming unassigned. Following on from EXAMPLE 5, page 3 can now be modified to start at PC1.1.20 by extending the range of page 2:

Example 6: **#MPD PC1.1.5>PC1.1.19=2**

PAGE DELETION

Page divisions are deleted by extending the range of the previous page to include both pages. Following on from EXAMPLE 6, page 2 can now be deleted by extending the range of Page 1 from PC1 to PC1.1.19:

Example 7: **#MPD PC1>PC1.1.19=1**

To cancel every page division all item designations are allocated to the first page.

Example 8: **#MPD PC1;999=1**

PAGE HEADLINES

If a MPD command causes one or more page divisions to be deleted, the user is requested to choose a new headline for this page. However, the user always has the option to update the headlines from one or more pages by adding the qualifier /H to his command line. To define Page 4 and allocate a headline:

Example 9: **#MPD PC1.1.25=4/H**

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.161 MPH - Modify Page Headline

Syntax: **MPH** [*pc_pgm*,]*page_nr_1*[>*page_nr_2*][=*page_nr_3*]

`MPH [pc_pgm,]page_nr_1[;number_of_pages][=page_nr_3]`

This command is available at PC level only.

Description: This command can be used to:

- Modify a headline for a single page,

- Modify the headlines for several pages where the range of pages can be specified either
  - From page_nr_1 to page_nr_2 or
  - From page_nr_1 for a number_of_pages.

In addition the headline from one page can be copied to one or more other pages by specifying =page-number_3. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Modify the headline for page 4 of PC3:

```
#MPH PC3,4
```

Example 2: Modify the headlines for pages 6 to 10 of PC2:

```
#MPH PC2,6>10 or #MPH PC2,6;5
```

Example 3: Copy the headline from page 1 to pages 2 to 10 of PC2:

```
#MPH 2>10=1
```

3.162 MPN - Modify Page Number

Syntax: **MPN** [*pc_pgm*,]*page_nr*[>*page_nr*]=*page_nr*[;*interv*]

MPN [*pc_pgm*,]*page_nr*[;*number_of_pages*]=*page_nr*[;*interv*]

MPN [*pc_pgm*,]*page_nr*=#

This command is available at PC level only.

Description: This command is used to modify the page number of existing pages for a PC_diagram. A single soft page break will be substituted by a hard page break. If you specify a range of page numbers, a hard page break must belong to the first and last page number in that range.
The third commandline is applicable to convert a hard page break to a soft page break.See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.163 MPRO - Modify PROperty

Syntax: **MPRO** *mlevref*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command modifies the property specified in the multi level reference. The value of the specified property will be listed and the new value can be supplied. If **<Enter>** is pressed then the old value will be preserved. Type HELP MLEVREF for the description of mlevref parameter.

3.164 MSTABS - Modify Symbol TABle Size

Syntax: **MSTABS**

This command is available at PC level only.

Description:

The MSTABS command allows the user to modify the size of the symbol table which is resident on the USER volume. (This will usually be the file PCMASS.SM). The MINIMUM, MAXIMUM and CURRENT sizes of the symbol table are displayed and the CURRENT size is taken as default. The MINIMUM size is the size of the symbol table already used and the MAXIMUM size is the size of the remaining USER volume space. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.165 MV - Modify Value

Syntax: **MV** [*pc_pgm*,]*variable name*

```
MV item_des:terminal
MV item_des:call parameter
```

This command is available at PC level only.

Description: This command permits the reading and in some cases, changing of the value of:

- A PC variable.
- An operational parameter.
- A constant.
- A call parameter.

The value may be changed if it is an operational parameter and in some cases a call parameter, indicated by an “=” in the dialog. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Call the command with a terminal identity:

```
#MV PC1.1.1:5
```

Example 2: Call the command with a variable name:

```
#MV PC1,signal
```

Example 3: Call the command in order to change a cycle time call parameter:

```
#MV PC1.1:C1
```

3.166 MVD - MoVe Debug point

Syntax: **MVD** *debug_no*,*addr*

```
MVD debug_no,statement
```

WARNING
This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description:

This command moves a debug point from its current address or statement to another within the same task. “debug_no” is debug point to be moved. “addr” is the address to which it is to be moved. This is valid only for debug points defined at addresses. “statement” is the statement to which the debug point is to be moved. This is valid only for debug points defined at statements.

Example: To move PASCAL Break point 4 to statement 136:

```
MVD 4,136
```

3.167 NEWPAGE

Syntax: **NEWPAGE**

This command is available at PC and system level.

Description: This command adds a page break to the printer output.

3.168 NEWDOC

Syntax: **NEWDOC**

This command is available at PC and system level.

Description: This command prints all pages collected in the printer buffer on the selected MS-Windows printer. After printing the buffer is cleared.

3.169 NOON

Syntax: **NOON** *error_code*

This command is available at PC and system level.

Description: This command removes the special action defined with the command “ON” and replaces it with the default action. See Section 6.3.2, Special Procedures, for more information.

3.170 O - Open

Syntax: **O** [*data_type*,]*addr*

WARNING
This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: The command is used to open a memory location i.e. its present value is printed out after which the system awaits a new value and/or a request to open the next or the preceding location or to terminate the command.

>	Opens the next memory location
<	Opens the preceding memory location
!	Terminates the command

Activate the last entered subcommand (>, <, !) by pressing <ENTER>. When the execution of an OPEN command begins, termination (!) applies. If data_type is omitted in a command, the latest entered applies. When the operator sets up a session with a target system, the data is set to UWH. Type HELP DATA_TYPE for description of data types.

Examples To open a word at address H’AAF0:

```
O UWH,H’AAF0
```

Above command causes the following display:

```
H’00AAF0 UWH = H’2317 =
```

Now, to alter value to H’90 and open H’AAF2, type:

```
90>
```

Or, to alter value to decimal 12 and open H’AAEE, type:

```
D’12<
```

Or, to alter value to H’720 and end the command, type:

```
720!
```

Or, to open H’AAF2 without changing value, type:

```
>
```

Or, to open H’AAEE without changing value, type:

```
<
```

Or, to end command without changing current value, type:

```
!
```

3.171 OBIT - Open BIT

Syntax: **OBIT** *addr*:*bit_number*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: Used to open a bit i.e. its current value is printed out after which the system awaits a new value and/or a request to open the next or the preceding bit or to terminate the command.

>	Opens the next bit
<	Opens the preceding bit
!	Terminates the command

Activate the last entered subcommand (>, <, !) by pressing <ENTER>. The command is available at byte level only; bit numbers 0 to 7 are permitted. It is possible to step past this range and open the bits in the next or previous byte. Only 0 and 1 are permitted as input values.

Examples To open the 6th bit of a byte at address H’AAF0:
OBIT H’AAF0:6

Above command causes the following display:

```
H’00AAF0:6 = 0 =
```

Now, to set this bit to 0 and open next bit, type:

```
0>
```

Or, to set this bit to 0 and open bit number 5, type:

```
0<
```

Or, to set this bit to 0 and end the command, type:

```
0!
```

Or, to open bit 7 without changing current value, type:

```
>
```

Or, to open bit 5 without changing current value, type:

```
<
```

Or, to end command without changing current value, type:

```
!
```

3.172 OLFILE - Open LogFILE

Syntax: **OLFILE** *logfile*[/**APPEND**,**READ**,**WRITE**][;**NOOUTPUT**]

This command is available at PC and system level.

Description: This command requests the system to copy all input/output to the specified logfile. “Logfile” is a segment name. If “/APPEND” is specified the new logging is appended to an existing file. “/READ” means that the logfile should be opened for read (see RLFILE). “/APPEND” and “/READ” are mutually exclusive (“/WRITE” is default). “;NOOUTPUT” means that only the interactive user input will be recorded in the file. Therefore the recorded logfile can be used as a command file afterwards. (NOTE: Do not forget to complete the command file with “EXIT” at the end!). See also keyword LOG_FILE.

See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.173 ON

Syntax: **ON** *error_code* **THEN GOTO** *label*

This command is available at PC and system level.

Description: This command redirects execution to a label where it shall continue if the specified error occurs. See Section 6.3.2, Special Procedures, for more information.

3.174 OPEN - OPEN segment

Syntax: **OPEN** *volume*:*segment*

This command is available at system level only.

Description: This command opens a segment on a volume so that you can read from or write into it using the console communication commands. Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name. In response to the command the system displays a start address (virtual) and size of the segment opened.
An attempt to access outside this range of addresses will cause a bus error if the segment is continuous. If it is a sequential segment the segment will be extended as long as there is space on the volume. Example:

To open segment X.SA on volume SOUR:

```
OPEN SOUR:X.SA
```

This command is not available with the Engineering Board PU516 and the RTA board.

3.175 OPS - Open Program Status register

Syntax: **OPS** [*task_name*]

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the contents of the program status register of a task and modify it if required. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.

Examples: To open program status register of task DEFS003:

```
OPS DEFS003
```

Above command causes the following display:

DEFS003 S LS LO NE VS

XNZVC = 00101 =

Now, to change values of desired flags, type:

10010

3.176 OREG - Open REGister

Syntax: **OREG** [[*task_name*,]*data_type*,]*register*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: The command opens a register, i.e. its present contents are printed out after which the system awaits a new value and/or a request to open the next or the preceding register or alternatively, to terminate the command.

>	Opens the next register
<	Opens the preceding register
!	Terminates the command.

The sequence between the registers is D0-D7, A0-A7 and PC. Activate the last entered subcommand (>, <, !) by pressing <ENTER>. If “data_type” is omitted in a command, the latest entered applies. Only the part of the register referred by data_type is affected.

Examples: To open register D2 of task DEFS003:

```
OREG DEFS003,ULH,D2
```

Above command causes the following display:

```
DEFS003 REG D2 ULH = H’0000A1FE =
```

Now, to alter value to H’FFFFA3 and open register D3, type:

```
H’FFFFA3>
```

Or, to alter value to H’ABCDEF and open register D1, type:

```
H’ABCDEF<
```

Or, to alter value to H’FEDCBA and end the command, type:

```
H’FEDCBA!
```

3.177 OTC - Open Task Code

Syntax: **OTC** [[*task_name*,][*data_type*],]*addr*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities and only available at system level.

Description: This command is used to read the code part of a task and modify it, if required. It is identical to the command **O** with three exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task’s code instead of absolute memory locations.
3. Default data type is always CODE.

Examples: To open the first word of the code of the task DEFS003:

```
OTC DEFS003,UWH,H’0
```

Above command causes the following display:

```
TC DEFS003 H’000000 = H’10EF =
```

Now, to alter value to H’FFFF and open next word, type:

```
H’FFFF>
```

Or, to open next word without modifying value, type:

```
>
```

Or, to alter value to H’EF09 and end command, type:

```
H’EF09!
```

3.178 OTDB - Open Task Data relative Base level

Syntax: **OTDB** [[*task_name*,][*data_type*],]*addr*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the data area of a task and modify it, if required. It is identical to the command O with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task’s global variables area i.e. relative register A5, instead of an absolute memory address.

Examples: To open a global integer located at offset -16 in task DEFS003:

```
OTDB DEFS003,SWD,-16
```

Above command causes the following display:

```
TDB DEFS003 -16 = 2079 =
```

Now, to modify value to 74 and open next word, type:

```
74>
```

Or, to open next word without modifying value, type:

```
>
```

Or, to modify value to 0 and end command, type:

```
0!
```

3.179 OTDC - Open Task Data relative Current level

Syntax: **OTDC** [[*task_name*,][*data_type*],]*addr*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the data area of a task and modify it, if required. It is identical to the command O with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the current Procedure ‘s local variables area i.e. relative register A6, instead of an absolute memory address

Examples:

To open a local integer located at offset -16 in task DEFS003:

```
OTDC DEFS003,SWD,-16
```

Above command causes the following display:

```
TDC DEFS003 -16 = 2079 =
```

Now, to modify value to 74 and open next word, type:

```
74>
```

Or, to open next word without modifying value, type:

```
>
```

Or, to change value to 0 and end the command, type:

```
0!
```

3.180 PD - Proceed from Debug point

Syntax: **PD** *debug_no*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command causes a task, which has been blocked at a debug point, to continue execution. “debug_no” is debug point where task is blocked (0 - 7).

Example: To proceed from debug point 4:

```
PD 4
```

3.181 PGPLC - ProGram PLC unit

Syntax: **PGPLC**

This command is available at PC and system level.

Description: The PGPLC command is used for programming a PLC unit connected to Advant Controller 400 Series. This command works with menus and a functional keyboard (keypad). Each menu is designed as the functional keyboard to make it easy to find the correct key. To select one function in a menu press the corresponding key on the keypad keyboard.

This command gives the possibility of getting help in every menu by pressing the **<ENTER>** key.

3.182 POPDB - Data Base POPulation

Syntax: **POPDB**

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command is used to modify and inspect the contents of the target data base.

The user maintains a dialog with the command and is guided by questions through the whole session. The user must be somewhat familiar with the logical structure of the data base and its items, like data dictionary, concept, logical file, instance, variant, property, group, format etc.

The data base identities can be referenced by name (symbolic) and/or by number (numerical). If there is no data dictionary, only numerical mode is available.

DIRECTION CHANGING COMMANDS

As a complementary part of the dialog there are three special commands:

!	Return to the above level.
>	Next property. Only on value level.
<	Previous property. On value level only.

Furthermore, a number of subcommands are available: *LIST, *L_ALL, *PRINT, *P_ALL, *FORM, *SHORT, *LONG, *DINS, *DMEMB, *H_MOD.

POPDB SUBCOMMANDS

Subcommands may be entered at any level of the dialog. A subcommand name starts with an “*”. The commands are listed below with their names in long and a short forms, their existing parameter alternatives and a short explanation.

*LIST *L	List data base contents for current level.
*L_ALL *LA	List all properties (even dynamic) for current level. Same usage as *LIST.
*PRINT *P	Print data base contents for current level.
*P_ALL*PA	Analogous to *L_ALL.
*LONG *LM	Set property listings in long mode.

*SHORT *SM	Set property listings in short mode.
*FORM *F	File name List format for concept on filename.SA.
*RUN *R	File name Execute a command file. (“batch mode”).
*DINS *DI	Instance_name or logical_file,logical_record Delete an instance from the data base.Care should be taken that the instance to be deleted is not referenced by another instance in the data base (external references).
*DMEMB *DM	Property_name or _number,member_number Delete a member from a repeating group. Even single formats can be removed.
*H_MOD *HM	Index;new_H_string Modify a part of a hexadecimal string, starting at the position “index, to the value of “new_H_string.

3.183 PROM - set up a PROM session

Syntax:
PROM
PROM *programmer*
PROM *programmer*;[*options*]
PROM *programmer*/*volume*:*segment*
PROM *programmer*/*volume*:*segment*;[*options*]
This command is available at system level only.

Description: This command sets up a session with a PROM programmer. At present only “PROMAC2” PROM programmers with 256Kbit, 512Kbit or 4MBit buffers are available (use "PROMAC2" also for the successor of Promac 2A, ANDO AF-9706.). If a segment identity is not supplied, it will be asked for later. If a segment identity is supplied, the segment is used to steer programming and provide necessary parameters. When the session is set up, a number of subcommands are available. A subcommand can be aborted pressing **<Ctrl-Break>**.

This command is not available with the Engineering Board PU516 and the RTA board.

Options and parameters

E	Do not print error messages.
I	Do not print information messages.
P	The segment supplied is of Program Image forma.
D	The segment supplied is of Data Dump format.
A	The segment supplied is of DUAP dump format.

Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Example 1: Segment at Program Image format supplied and do not print information messages:
PROM PROMAC2_256/VOLA:SEGA.PI;IP

Example 2:No segment identity supplied:
PROM
This command is not available with the Engineering Board PU516 and the RTA board.

3.184 R - Read

Syntax: **R** [*data_type*,]*addr*
R [*data_type*,]*start_addr*>*end_addr*
R [*data_type*,]*start_addr*;*number*
This command is part of the product Expert Utilities.
This command is available at system level only.

Description: Can be used to read the contents of one or more absolute memory location(s). It has three forms.

1. Read one unit of data from memory location “addr”.
2. Read as many units of data as available in the range “start_addr” to “end_addr”.
3. Read “number” units of data (<32767) starting from location “start_addr”.

“data_type” specifies the output format as well as the number of bytes forming a unit of data. If data type is omitted, the last type entered is assumed. Type HELP DATA_TYPE for description of data types.

Example 1: To read an Unsigned Word in Decimal notation from address H’AAF0:
R UWD,H’AAF0

Example 2: To read range H’400 to H’410 as ASCII characters:
R C,H’400>H’410

Example 3: To read 14 Signed Words in Hexadecimal notation from H’100:
`R SWH,H’100;14`

3.185 RDATIME - Read DAte and TIME

Syntax: **RDATIME**

This command is available at PC and system level.

Description: This command displays system date and time as year-month-day/hour:minute:second. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.186 RECONFIG - REstart CONFIGuration

Syntax: **RECONFIG**

This command is available at PC and system level.

Description: The **RECONFIG** command corresponds to “cold start” (START MODE switch set to AUTO) of the system and requires configuration mode.

You are asked whether all PC programs and the data base are to be removed. The **RECONFIG** command deletes the entire application programs in the connected Advant Controller 400 Series. But the **RECONFIG** command does not delete the corresponding data on the USER volume. For this change to SYSTEM level by using the **SLLEV SYST** command and delete the files: *.SM, *.AP, *.UI, if present, to avoid mismatching of the USER volume with the target system. The result of this procedure is an initial USER volume. Perform a **SLLEV PC** command directly afterwards.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.187 RECV - RECeiVe

Syntax: **RECV**

```
RECV volume:segment,segment_stor,target,target_stor,size,option
```

This command is available at system level only.

Description: This command receives a segment from the target system and writes it on a volume. If you issue the command by typing just RECV (Syntax 1) the parameters are accepted in an interactive mode.

Parameters Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

segment_storageIs type of output (ASC,BI,DD,LO,PI or RO).

targetIs sending computer (MA7 or DS).

target_storageIs transmission format (ASC,BI,LO,PI or RO).

sizeIs size of the segment in blocks of 1 kbytes.

optionIs L (List), Q (Query), S (Suppress) or N (reuse segment).

Example: To receive segment as XYZ.LO of size 120 blocks on volume MA01 from MA700:
RECV MA01:XYZ.LO,LO,MA7,BI,120,L

With the Engineering Board PU516 and the RTA board, this command does not support communication with VAX computers.

3.188 RENAME - RENAME segment

Syntax: **RENAME** [*volume*]:*old_name*>*new_name*[;*prot*]

This command is available at PC and system level.

Description: This command alters the name and/or privilege code of a segment on a volume. If “volume” is omitted previously specified name is assumed.

old_nameIs name of segment to be renamed.

new_nameIs the new segment name.

protIs privilege code (R for Read; RW for Read and Write).

Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

Example: To rename XAAP:MY.SA to XAAP:YOUR.SA with Read privilege:
RENAME XAAP:MY.SA>YOUR.SA;R

3.189 RESTART - load prommed application and RESTART system

Syntax: **RESTART**
This command is available at PC and system level.

Description: This command loads prommed backup and restarts target system.

3.190 RET - REsume Task

Syntax: **RET** *task_name,reason*

WARNING

This command shall be used by experts only!

This command is available at system level only.

Description: This command resumes a task with a resume reason. Type HELP TASK_NAME for description of task names. Parameter “reason” is the resume reasons of the task (an integer). A task can be resumed only when it has been suspended with the same resume reason.

Example: To resume task DEFS003 with reason = 3:
RET DEFS003,3

3.191 RETURN

Syntax: **RETURN**

This command is available at PC and system level.

Description: This command passes execution control back to the calling command file, at the line after the subroutine call (“EXECSUB”). See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.192 RLFILE - Read LogFILE

Syntax: **RLFILE** [*line*,]*symbol*[/**GLOBAL**]

This command is available at PC and system level.

Description: This command reads a line from the logfile and assigns it as a text string to a given symbol (default local). The logfile must have been opened for read (see OLFILE). First line in the file is 1 (default). See also keyword LOG_FILE. Note that lines beginning with a command are skipped.

3.193 RLMD - Read Local Motor Drive

Syntax: **RLMD** [*board_addr*,][*data_type*,]*addr*[>*addr*]

`RLMD` `[`*`board_addr`*`,][`*`data_type`*`,]`*`addr`*`[;`*`number`*`]`

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: RLMD is used for reading the circuit board memory on DSDC 110 (Digital speed control).

If board_addr and data_type are omitted the latest entered applies.

Example 1: Read one hexadecimal byte on address H’6800 on a board with I/O address H’FFE000.

```
*RLMD H’FFE000,UBH,H’6800
```

Example 2: Read eight boolean from address H’6800 on a board with I/O address H’FFE000.Boolean is last entered data type.

```
*RLMD H’FFE000,,H’6800;8
```

Example 3: Read from address H’6800 to address H’6900 and use the same board_address and data_type as last time:

```
*RLMD H’6800>H’6900
```

3.194 RMD - Remove Display

Syntax: **RMD**
This command is available at PC level only.

Description: This command terminates the display of the table with variable status. The display then returns to a single field presentation. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.195 RREG - Read REGisters

Syntax: **RREG** [*task_name*]
This command is part of the product Expert Utilities.
This command is available at system level only.

Description: This command is used to read the contents of the address and data registers of a task. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.

Example: To read registers of the task DEFS003:
RREG DEFS003

3.196 RSIF - ReSet Inhibit Flag

Syntax: **RSIF** *PC_program_ n*[>*PC_program_m*]

DANGER

Danger of personal risk! Risk of malfunction!

This command shall be used by experts only!

This command is available at PC level only.

This command resets the inhibit flag in one PC program or a range of PC programs, which were set whith the **SEIF** command.

See also: **LIF**, **SEIF**, **SWBL**.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.197 RTC - Read Task Code

Syntax: **RTC** [[*task_name*,] [*data_type*],] *addr*

```
RTC [[task_name,] [data_type],] start_addr > end_addr
RTC [[task_name,] [data_type],] start_addr ; number
```

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the code part of a task. It is identical to the command R with three exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task's code instead of an absolute memory address.
3. Default data type is always CODE.

Example 1: To read the first 20 words of the code of task CXAP0010:

```
RTC CXAP0010,UWH,H'0;20
```

Example 2: To read same code as CODE (task_name defaulted):

```
RTC H'0;20
```

3.198 RTDB - Read Task Data relative Base level

Syntax: **RTDB** [[*task_name*,][*data_type*],]*addr*

```
RTDB [[task_name,][data_type],]start_addr>end_addr
RTDB [[task_name,][data_type],]start_addr;number
```

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the data part of a task. It is identical to the command R with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task's global variable area i.e. relative register A5. It can be positive or negative unlike absolute addresses.

Example 1: To read a global real variable at address offset -210:

```
RTDB DEFS003,R,-210
```

Example 2: To read a character array of size 12 starting at offset -108:

```
RTDB DEFS003,C,-108;12
```

3.199 RTDC - Read Task Data relative Current level

Syntax: **RTDC** [[*task_name*,][*data_type*],]*addr*

```
RTDC [[task_name,][data_type],]start_addr>end_addr
RTDC [[task_name,][data_type],]start_addr;number
```

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to read the data part of a task. It is identical to the command R with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the current Procedure's local variable area i.e. relative register A6. It can be positive or negative unlike absolute addresses.

Example 1: To read a local real variable at local address offset -10:

```
RTDC DEFS003,R,-10
```

Example 2: To read local character array of size 12 starting at offset -10:

```
RTDC DEFS003,C,-10;12
```

3.200 RTIME - Read TIME

Syntax: **RTIME**

This command is available at PC and system level.

Description: This command displays system time as hour:minute:second. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.201 RUDAS - Retrieve User Disk Application Segments

Syntax: **RUDAS**

This command is available at PC level only.

Description: This command copies the application segments, which are saved in the backup area in the target system, to the USER volume. The old application segments on the volume will be deleted. This has to be confirmed, otherwise the copying will not be executed. On message “File busy”, go to SYSTEM level and close the file, return to PC level and type command again.
See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.202 SDATIME - Set DAte and TIME

Syntax: **SDATIME** *year-month-day*/*hour*:[*minute*]:[*second*]

This command is available at PC and system level.

Description: This command sets the system date and time.

year	Is year (a 4 digit integer like 1981).
month	Is month (1 - 12).
day	Is date (1 - 31).
hour	Is hours (0 - 23).
minute	Is minutes (0 - 59).
second	Is seconds (0 - 59).

If either “minute” or “second” is not specified value 0 is defaulted. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: To set date and time without defaulting any value:

```
SDATIME 1982-08-20/0:47:0
```

Example 2: To set date and time by defaulting value for seconds:

```
SDATIME 1982-01-23/10:20:
```

3.203 SEIF - SEt Inhibit Flag

Syntax: **SEIF** *PC_program_ n***[>***PC_program_m*]

DANGER

Danger of personal risk! Risk of malfunction!

This command shall be used by experts only!

This command is available at PC level only.

This command sets the inhibit flag in one PC program or a range of PC programs. If the inhibit flag is set, it prevents resetting of the corresponding data base element output values during execution of an **EBM** or **DIBM** command. Without inhibit flag those values would be set to zero.

See also: **LIF**, **RSIF**, **SWBL**.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.204 SELNM - SELect Network Management

Syntax: **SELNM**

This command is available at system level only.

Description: The SELNM command can be used to activate the Network Management. The SELNM command transfers the execution level from the SYSTEM level to Network Management level. This level (NM) is indicated by the prompt NM>

Example: ***SELNM**
NM>

3.205 SEND

Syntax: **SEND**
SEND *volume*:*segment*,*segment_stor*,*target*,*target_stor*,*option_list*
This command is available at system level only.

Description: This command sends a segment from a volume to a target computer. If the command is issued by typing just SEND (syntax 1) the parameters are accepted in an interactive mode.

Parameters Type HELP VOLUME_NAME for description of volume name. Type HELP SEGMENT_NAME for description of segment name.

segment_stor	Segment type(ASC, BI, DD, LO, PI or RO).
target	Receiving computer (MA7 or DS).
target_stor	Transmission format (ASC, BI, LO,PI or RO).
option_list	L (List), Q (Query) or S (Suppress).

Example:To send segment XYZ.DD from volume MA01 to Advant Controller 400 Series:
SEND MA01:XYZ.DD,DD,DS,BI,L
With the Engineering Board PU516 and the RTA board, this command does not support communication with VAX computers.

3.206 SEOM - SEt Output Medium

Syntax: **SEOM** [**C**[,**P**]]
SEOM [**P**[,**C**]]
This command is available at PC and system level.

Description: This command enables the user to select either the console or the printer, or both as the output medium. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: Enable only the console for output:
#SEOM C

Example 2: Enable only the printer for output:
#SEOM P

Example 3: Enable both of them:
#SEOM C,P

3.207 SEPS - SEt Page Size

Syntax: **SEPS**
This command is available at PC and system level.

Description: This command opens the **Page Setup** dialog. The dialog is described in the AdvaBuild On-line Builder User’s Guide ($$ for engineering stations or for the OS too?).

3.208 SERN - SEt Real Notation

Syntax: **SERN**
This command is available at PC level only.

Description: This command is used to define the notation for values of data type real. The notation can be exponential (e.g. 3.15E4) or non-exponential (e.g 31500.00). See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.209 SHLSYMB - SHow Local SYMBol

Syntax: **SHLSYMB** [*symbol*]
This command is available at PC and system level.

Description: This command lists the value of the specified local symbol. The symbol is first searched in current levels' symbol table, then in upper level and so on. If the name is omitted, values of all the local symbols are listed. If there is no local symbol with the given name, values of all the local symbols with names starting with the given name are listed. symbol : name of symbol

See also help on command SHSYMB and keyword SYMBOLS. See Section 6.5.4, Displaying Symbol Values, for more information.

Example (2): **SHLSYMB TEXT_I**
SHLSYMB
In example 1 symbol(s) with name TEXT_I is listed. Example 2 shows how one can list all the symbols from the table.

3.210 SHSYMB - SHow (global) SYMBol

Syntax: **SHSYMB** [*symbol*]
This command is available at PC and system level.

Description: This command lists the value of the specified global symbol. If the name is omitted, values of all the global symbols are listed. If there is no global symbol with the given name, values of all the global symbols with names starting with the given name are listed.
symbol : name of symbol. See also help on command SHLSYMB and keyword SYMBOLS. See Section 6.5.4, Displaying Symbol Values, for more information.

Example (2): **SHSYMB PC_1**
SHSYMB PC

In example 1 symbol with name PC_1 is listed. Example 2 shows how one can list the same symbol and also all other symbols with names starting with PC.

3.211 SHTARG - SHow TARGet

Syntax: **SHTARG**

This command is available at PC and system level.

Description: This command prints the system identification string for current target system.

3.212 SHUDAS - SHow User Disk Application Segments

Syntax: **SHUDAS**

```
SHUDAS /TARG
SHUDAS /USER
```

This command is available at PC and system level.

Description: /TARGLists all segments stored in backup area.

/USER	Lists all USER disk application segments located on the actual USER volume of the engineering station.

This command displays the version number and the unique identity for the application segments that are stored on USER volume and in the backup area in the target system. The time when the application segments were stored in the target will be shown, too. The volume name and the segment name of the USER volume stored in the backup area in the target will also be displayed, if they differ from the default names (default USER PCMASS). Corrupt USER volume or backup area will be indicated.

Two different groups of USER disk application segments exist:

*.SM, *.AP, P1PCMASS.NI:

These segments will be modified during a session.

*.TB, *.TC, *.TI :

These segments will not be modified during a session. They belong to the source code created by the Function Chart builder. They are used to identify type circuits during backtranslation of source code within the Function Chart Builder. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.213 SLG - SeLect Graphical mode

Syntax: **SLG**

This command is available at PC level only.

Description: This command sets the output mode selector to graphical mode. SLG is default for the mode selector. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.214 SLLEV - SeLect command LEVel

Syntax: **SLLEV** *level*

This command is available at PC and system level.

Description: When you work on a process controller two command levels are available:

- PC level (#).
- System level (*).

The PC level covers application programming work, whereas the system level corresponds more to system specific work. A ‘*’ appears on the screen if the system level is selected. A’#’ stands for the PC level. Use the SLLEV command to switch between the two levels:

Only those commands are available you need to work on one of the levels. Nevertheless some commands are available on both levels.

Example (2): ***SLLEV PC**

Starts the PC level
#SLLEV SYST

Starts the system level

3.215 SLNG - SeLect Non-Graphical mode

Syntax: **SLNG**

This command is available at PC level only.

Description: This command sets the output mode selector to non-graphical mode. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.216 SLNUM - SeLect NUMeric identities

Syntax: **SLNUM**

This command is available at PC level only.

Description: This command sets the output mode selector to numeric identities. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.217 SLSYM - SeLect SYMbolic identities

Syntax: **SLSYM**

This command is available at PC level only.

Description: This command sets the output mode selector to symbolic identities. SLSYM is default for the mode selector. See *MasterPiece 200/1 Configuration and Programming* manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.218 SLTARG - SeLect TARGet

Syntax: **SLTARG** [*network*,*node*]
This command is available at PC and system level.

Description: This command terminates current session and sets up a new one according to the parameters. There is three possible combinations for the parameters:

1. If network/node is omitted, then a session is set up with the target system that the engineering station is physically connected to.
2. Network/node = 0 means that a local session (With the engineering station itself) is set up.
3. If both network/node are in the range 1 - 127, a session with a remote network is assumed.

For more information see Section 5.5, Targets in a Network Environment.

3.219 STIME - Set TIME

Syntax: **STIME** *hour*:[*minute*]:[*second*]
This command is available at PC and system level.

Description: This command sets the system time.

hourIs hours (0 to 23).

minuteIs minutes (0 to 59).

secondIs seconds (0 to 59).

If either “minute” or “second” is not specified value 0 is defaulted. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example 1: To set time without defaulting any value:

STIME 7:13:24

Example 2: To set time by defaulting value for minutes:
STIME 12::10

3.220 SUDAS - Save User Disk Application Segments

Syntax: **SUDAS**
This command is available at PC level only.

Description: This command copies the application segments from the USER volume to the backup area in the target system, i.e. it updates the copy of the application segments in the target system.
On message “File busy”, go to SYSTEM level and close the file, return to PC level and type command again. See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

NOTE

Do not use older ONB versions (prior to ONB*2.4/0) to save the application segments in target if the application contains instance names! The segments will be lost !

3.221 SUSPEND

Syntax: **SUSPEND**
This command is available at PC and system level.

Description: This command suspends the console dialog to let other printouts (e.g. system messages) come out. Press **<Ctrl-Break>** to resume the console dialog.

3.222 SWBL - SWitch BLockings

Syntax: **SWBL** [**DB,**] *PC_Program_A* / *PC_Program_B*

DANGER

Danger of personal risk! Risk of malfunction!

This command shall be used by experts only!

Switching the executing PC program may affect the process as soon as the change is carried out. Do not switch the execution from one PC program to another one, unless you are familiar how the process will be affected by the change.

This command is available at PC level only.

Description: This command can be used to block all execution units or ouput to the data base of a PC program and de-block all execution units of another PC program. If the optional parameter **DB** is omitted, the running PC Programm will stop, while the other one will start running. If you specify **DB**, the running program will keep running, but the output to the data base and therefore to the real process will be blocked.

You can inspect the status of the involved PC programs with the **LSS** command (List Special Status).

The PC programs in the **SWBL** command can be given in an abritary order. The command checks which of the PC programs is blocked and which is running.

See also: **LIF**, **RSIF**, **SEIF**.

See *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

3.223 TRDBS - TRanslate Data Base Source

Syntax: **TRDBS** [*volume*[:*segment*[;*option*[,*noferr*][,**IEB**][,**IV**]]]]
This command is available at PC and system level.

Description:

volume	Volume name, 4 characters.
segment	Segment name, max. 8 characters.
option	Short or long mode to disk or printer (**S**, **L**, **SP**, **LP**).
noferr	Maximum number of errors before translation is stopped.
IEB	An insert error stops the translation.
IV	Ignore volume name given in source code file.

Loads and translates the data base source code from the specified volume and segment name.

NOTE

- TRDBS is an interactive command and will prompt the user for all required input, including the initial parameters (if omitted).
- Input can be generated by the command: DUDBS - (DUmp Data Base Source) or created by the AdvaBuild Function Chart Builder or with a text editor.
- If both volume and segment names are specified, but no options, default options are used, i.e. short mode and error file on disk.
- Command will update symbols indicating how the command performed. The symbols are:
  - MA$STATUS_CMD : Name of command, i.e. TRDBS.
  - MA$STATUS_CODE : Error code, where zero (0) indicates success and one (1) that break was input. Other values indicate first error.
  - MA$STATUS_TEXT : Text describing the error first encountered (If any).

The symbols can be tested e.g. in command files. See *ABB Master Source Code Handling*.

Example 1: Translate source code file in short mode with error file on disk.

```
#TRDBS ABCD:DB01
```

or

```
#TRDBS ABCD:DB01;S
```

Example 2: Translate source code file in long mode with error file on printer. Break after 12 errors.

```
#TRDBS ABCD:DB02;LP,12
```

Example 3: Translate source code file in short mode with error file on printer. Break if any insertion fails.
#TRDBS ABCD:DB03;SP,IEB

Example 4: Translate source code file in long mode and ignore the volume names within source code file. Break after 12 errors.
#TRDBS ABCD:DB02;L,12,IV

3.224 TRPCS - TRanslate PC Source

Syntax: **TRPCS** [*volume*[:*segment*[;*option*[,*noferr*][,**ISB**][,**IV**]]]]

This command is available at PC level only.

Description: volume : Volume name, 4 characters, default is SRCE

segment :	Segment name, max. 8 characters
option :	Short or long mode to disk or to printer (**S**, **L**, **SP**, **LP**)
noferr :	Maximum number of errors before translation is stopped
ISB:	An error during IS stops the translation
IV:	Ignore volume ID given within source code files

This command is used to translate source code on mass memory to object code (PC language) in the target system. When the command is started TRPCS with only volume as parameter prompts for each source code segment found on specified volume. It is possible to select one or more segments to be translated.

NOTE

The command will update symbols indicating how the command performed. These symbols are:

MA$STATUS_CMD : Name of the command, i.e. TRPCS.

MA$STATUS_CODE : Error code, where zero (0) indicates success and one (1) that break was input. Other values indicate first error.

MA$STATUS_TEXT : Text describing the error first encountered (If any).

These symbols can be tested for example in command files.

See *ABB Master Source Code Handling* manual for more information.

3.225 TSESS - Terminate SESSion

Syntax: **TSESS**

This command is available at PC and system level.

Description: This command terminates current session.

3.226 TYPE

Syntax: **TYPE** *text*

This command is available at PC and system level.

Description: This command prints the specified message. The text to be printed must be enclosed in double quotes. See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.227 UHR - Update Harddisk Remotedirectory files

Syntax: **UHR**

This command is available at PC and system level.

Description: For internal use on the engineering station only! To update directory files after copying of files on operating system level.

This command is not available with the Engineering Board PU516 and the RTA board.

3.228 UNCSSD - UNCompresS System Data

Syntax: **UNCSSD**

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command unpacks a compressed system data area. The data is put in the RWM area, which must have been cleared before the command is given. The target can only be an “off-line” system and the session must have been set up “non-symbolic”. Type HELP INSTALL for more information.

This command is not available with the Engineering Board PU516 and the RTA board.

3.229 UNLOCK

Syntax: **UNLOCK** *pc_pgm*

This command is available at PC level only.

Description: This command can be used to UNLOCK a PC program that has been locked with the LOCK command.

See *MasterPiece 200/1 Configuration and Programmin*g manual or *AMPL Configuration Advant Controller 400 Series Reference Manual* for more information.

Example: **#UNLOCK PC1**

```
Password: ___________
```

3.230 USERINPUT

Syntax: **USERINPUT**

This command is available at PC and system level.

Description: This command passes execution control from the command file to the terminal (the command file is resumed with “CONTINUE”). See Section 6.1.3.1, Commands to Control Input and Output, for more information.

3.231 VERIFY

Syntax: **VERIFY ON|OFF**

This command is available at PC and system level.

Description: This command controls whether lines in command files are displayed at the terminal and, if present, copied to a logfile. See Section 6.1.3.3, Commands to Control the Execution Flow, for more information.

3.232 W - Write

Syntax: **W** [*data_type*,]*addr*=*value*

```
W [data_type,]start_addr>end_addr=value
W [data_type,]start_addr;number=value
```

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to write into the memory. It has three forms:

1. Write a unit of data at memory location specified by “addr”.
2. Fill the range “start_addr” to “end_addr” with the same data unit.
3. Write the same unit of data “number” times (<32767) starting from location “start_addr”.

Type HELP DATA_TYPE for description of data types. If data type is omitted, the last type entered is assumed. “value” specifies the data to be written.

Example 1: To write decimal value 2 in word at address H’AAF0:

```
W SWD,H’AAF0 = 2
```

Example 2: To write char “x” in range H’400 to H’410:

```
W C,H’400>H’410 = ‘x’
```

Example 3: To write H’F at 14 bytes starting from H’100:

```
W UBH,H’100;14 = H’F
```

3.233 WLFILE

Syntax: **WLFILE** *text*

This command is available at PC and system level.

Description: This command writes a text line to the logfile (opened with OLFILE). The text to be printed must be enclosed in double quotes. The command may only be used from within a command file.

3.234 WTC - Write Task Code

Syntax: **WTC** [[*task_name*,][*data_type*],]*addr*=*value*
WTC [[*task_name*,][*data_type*],]*start_addr*>*end_addr*=*value*
WTC [[*task_name*,][*data_type*],]*start_addr*;*number*=*value*

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to write in the code part of a task. It is identical to the command W with three exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task's code instead of an absolute memory address.
3. Default data type is always CODE.

Example 1: To write H'0 in first 20 words of task code DEFS003:

```
WTC DEFS003,UWH,H'0;20 = H'0
```

Example 2: To write NOP instructions instead (data_type defaults to CODE, task_name to previous task_name):

```
WTC H'0;20 = NOP
```

3.235 WTDB - Write Task Data relative Base level

Syntax: **WTDB** [[*task_name*,][*data_type*],]*addr*=*value*

```
WTDB [[task_name,][data_type],]start_addr>end_addr=value
WTDB [[task_name,][data_type],]start_addr;number=value
```

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to write in the data part of a task. It is identical to the command W with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the given task's global variable area i.e. relative register A5. It can be positive or negative unlike absolute addresses.

Example 1: To assign value 10.0 to global real at offset -210 in task DEFS003:

```
WTDB DEFS003,R,-210 = 10.0
```

Example 2: To assign value “xxxx” to a character array at offset -108:

```
WTDB DEFS003,C,-108;4 = 'x'
```

3.236 WTDC - Write Task Data relative Current level

Syntax: **WTDC** [[*task_name*,][*data_type*],]*addr*=*value*

```
WTDC [[task_name,][data_type],]start_addr>end_addr=value
WTDC [[task_name,][data_type],]start_addr;number=value
```

WARNING

This command shall be used by experts only!

This command is part of the product Expert Utilities.

This command is available at system level only.

Description: This command is used to write in data part of a task. It is identical to the command W with two exceptions:

1. If task name is omitted, the last name entered is assumed. Type HELP TASK_NAME for description of task names.
2. The address specified is assumed to be an offset from the start address of the current Procedure ‘s local variable area i.e. relative register A6. It can be positive or negative unlike absolute addresses.

Chapter 4 Special Commands for HP-UX Environment

4.1 dobackup

Syntax: **`dobackup`** *project net1*,*node1* [*netn*,*noden*]

Description: Creates backups for a series of nodes using PUbackup.

Example: Backup of 1 node:

```
dobackup pump_1 12,2
```

Backup of 4 nodes:

```
dobackup pump_1 12,2 12,3 12,4 12,5
```

4.2 dorestore

Syntax: **`dorestore -p|-d`** *project*|**-n** *project date_directory*

`dorestore` *`project date_directory net`*,*`node`* [*`netn`*,*`noden`*]

Description: Restores a number of nodes from a backup created with dobackup.

Example: Restore of 1 node:

```
dorestore pump_1 931025-14:58 12,2
```

Restore of 4 nodes:

```
dorestore pump_1 931025-14:58 12,2 12,3 12,4 12,5
```

NOTE

Before restoring a backup, you must reconfigure a node.

4.3 PUbackup

Syntax: **`PUbackup`** network,node

Description: Creates a backup of the given target system (DUAP). The dump is stored on the volume SRCE, which must have been defined before starting the command.

4.4 PUboards

Syntax: `PUboards`

Description: Searches for available Real Time Accelerator (RTA) board(s). Writes the number of active boards to stdout.

4.5 PUlock

Syntax: `PUlock`

Description: Allocates a slave board, starts the mass memory task and defines the system volumes (SYST, CMDS).

4.6 PUrelease

Syntax: `PUrelease [-f]`

Description: Deallocates the Real Time Accelerator board and stops the associated tasks. The argument “-f” executes the command **TSESS**.

4.7 PUrestore

Syntax: `PUrestore` network,node

Description: Restores an application dump. The dump must haven been created with the PUbackup command. The volume SRCE must be defined according to the directory where the dump is stored. The argument “-r” executes the command **RECONFIG**, the argument “-o” switches the target node to the operation mode.

4.8 PUvoldef

Syntax:
`PUvoldef` [interactive version]
`PUvoldef` filename [read contence of file]
`PUvoldef` volume_number volume directory

Description: Defines the user specific volume number (0...7) and volume name (USER, SRCE).

4.9 rta_boot

Syntax: `rta_boot`

Description: Boots the Real Time Accelerator (RTA) board.

Chapter 5 Working Methods of the On-line Builder

5.1 Volumes and Segments

The On-line Builder has access on two volumes, SRCE and USER. On the engineering station a set of these two volumes exists in order to store any target specific data for each defined target. Depending on the target selection on the engineering station the corresponding set of volumes (SRCE and USER) are available and refer to the directories within the directory structure of the selected target.

It is recommended to use the SRCE volume, which is the default volume when working with the On-line Builder rather then the USER volume. The USER volume contains different segments with various application data for internal use of the On-line Builder.

Each volume may contain a number of segments. A segment is a certain quantity of data or program code which is grouped together. A segment is handled as a unit by the mass storage and has a segment name, which corresponds with a file on the according directory.

A segment has the following segment identifier, that contain a colon and a period as separators:

```
volume:segment.extension
```

The following characters are valid (other characters may appear in volume names given by the system):

volume	= 4 ASCII characters; A-Z, a-z, 0-9.
segment	= 1-8 ASCII characters; A-Z, a-z, 0-9.
extension	= 2 ASCII characters; A-Z, a-z, 0-9.

Example:

`USER:ABCD.AA`

In syntax definitions the extension of a segment is not mentioned, but is a part of “segment”.

5.2 Wildcards

A wildcard is a character which stands for any other character.

Wildcard characters are accepted by many commands when handling segment identifiers. Two wildcard characters are accepted:

%	(percent) means that any character can occur in this position.
*	(star) works in the same manner as %, but with the difference that the star applies for all subsequent positions. Applicable within a segment name or within an extension.

Example 1:

```
#LDIR USER:A*.%1
```

Lists the segments from USER volume which have segment names beginning with A, and extensions ending with 1.

Example 2:

```
#LDIR :A%%.*
```

Lists all segments whose segment name consists of three characters, the first of which is A. The extension can be optional.

5.3 Default Volume and Segment Names

Many commands accept default volume and segment names. However, separators between the parameters must always be written (":" and ">").

The volume and segment names are stored in variables called S1 and S2, which retain the names in the segment identifiers. A command which contains only one identifier uses S2. A command which contains two identifiers uses S1 for the first identifier and S2 for the second.

Each time an identifier is given a command, S2 is updated (as is S1, if two identifiers are given). In order to explain the function of S1 and S2, a few commands are used in the following example, which are described in detail later in this manual.

Example:

```
#COPY USER:S*.CP>SRCE:*.*
```

copies the segments from volume USER to volume SRCE, whose segment names begin with S and whose extensions are equal to "CP". The variable S1 gets the value USER and S2 SRCE.

The contents of both segments can be compared with:

```
#COMPARE : > :
```

The system responds with the message that the files are equal.

However, there is an exception to this:

```
#LDIR SRCE:
```

is synonymous with

```
#LDIR SRCE:*.*
```

which updates S1 instead of S2.

5.4 Levels in On-line Builder

The On-line Builder knows three different levels:

- System level (*).
- PC level (#).
- Disconnected level (?).

Each level is indicated by its specific prompt (*, #, ?) on the command line, which allow you to enter a command.

NOTE

When running the On-line Builder with the Engineering Board DSPU131, the disconnected level (?) is not available.

After starting a session the level can be changed between system level and PC level using the command **SLLEV** (see Section 3.214, SLLEV - SeLect command LEVel). The main difference between those two levels is a different selection of commands of the started session. The PC level supports you with application programming oriented commands. The system level stands for more system maintenance oriented commands. Various commands are available in both levels.

The disconnected level (Advant Station 500 Series only) can be activated with the command **DCTARG** (see Section 3.31, DCTARG - DisConnect TARGet). This means the engineering station will be disconnected but the command set of that session remains in the engineering station and need not to be reloaded when re-activating the session again.

5.5 Targets in a Network Environment

Selecting a Target

With the **SLTARG** (Select target) command you can remotely connect the engineering station to any target within the control network the engineering station is physically connected to. The **SLTARG** command gives a quick way to service systems included in a network. You can use this command interactively, but also in a command file.

Example:

```
#SLTARG 11,1
```

Syntax:

```
SLTARG [network,node]
```

Description:

Network	Denotes the network number.
Node	Denotes the node number.

There are three possible combinations for the command:

- If network/node is omitted, then a session is set up with the target system that the engineering station is physically connected to.
- Network/node = 0 means that a local session (with the engineering station itself) is set up (for special cases only).
- Finally, if network/node are both values in the range 1 - 127, a session with a remote network is assumed.

Showing a Target

Syntax:

```
SHTARG
```

Description:

The command prints the system identification string including network node number for current target system. With this command you can confirm that the engineering station is connected to the correct target system.

Chapter 6 Command Files

This chapter describes how to create and use command files for the engineering station.
A command file consists of a sequence of commands. Commands which are used often can simply be collected and executed as a command file. It is possible to construct command files with loops and conditional execution. Any key sequences can be recorded (command: **OLFILE** and argument: ;**NOOUTPUT**) and used as command file.

This chapter also explains user-defined symbols and how to create a user interface with menus in a command file.

6.1 Building a Command File

You may build up a command file of three different kind of components:

- User commands
- Comments
- Commands to control the execution.

6.1.1 User Commands

User commands are those commands, which are normally given directly from the keyboard.

NOTE

Commands, which work interactively, read data from subsequent lines in the file. They interpret them as data input or as sub commands. Therefore these lines must not contain any prompt.

You can record interactive application programming key sequences (command: **OLFILE** and argument: ;NOOUTPUT) and also use this key sequence for a command file.“;NOOUTPUT” means that only the interactive user input will be recorded in the file.

NOTE

Do not forget to complete the command file with the **EXIT** command at the end!

When using the command **OLFILE** the command **CLLFILE** is always needed to stop recording and to make the file available.

Examples

Example 1:

```
#!
#LPCD PC2
#!
```

In this example, the command **LPCD** (List PC Diagram) is called along with the name of the PC program (PC2) specified in the command file.

Example 2:

```
#!
#LPCD ’P1’
#!
```

In this example, the command is called with the PC program specified as the value for the symbol P1 (for information about symbols see Section 6.5, User-defined Symbols). P1 is the first parameter transferred to the command file. For example, P1 can be defined as the character string PC2.

6.1.2 Comments

Comments are used for clarification of the contents in a command file. A comment line must be preceded by an exclamation point (!).

The command lines which are executed in a command file are normally not displayed on the screen. Using the **VERIFY** command (described later in this section), the lines can be echoed and the execution followed on the screen. In this case, the comment lines which are not to be displayed must be preceded by two exclamation points (!!).

Example:

```
!
!Execution begins
!
LDIR SRCE:
!!
!!should an error message be given?
!!
LDIR USER:
```

6.1.3 Commands to Control the Execution

These commands are divided into three subgroups:

- Commands to control input and output.
- Symbols in command files.
- Commands to control the execution flow.

6.1.3.1 Commands to Control Input and Output

TYPE

The **TYPE** command writes a specified text on the screen. It is used to present messages to the operator.

Syntax: **TYPE** text

Description: This command prints the specified message. The text to be printed must be enclosed in double quotes. This parameter follows the rules for text strings described in Section 6.5.6.1, Character Strings.

Example:

```
!
TYPE "Execution begins"
TYPE ""
LDIR SRCE:
!
```

USERINPUT

The **USERINPUT** command orders the engineering station to read commands from the keyboard rather than from the command file.

Syntax: **`USERINPUT`**

Description: The command file stops execution and waits for input. When the input is given, enter the command **CONTINUE**. The input returns then to the command file, which continues. Use the **TYPE** command to inform the user about the required input.

Example:

```
!
TYPE "Investigate contents of diskette!"
USERINPUT
```

The execution of the command file stops here. You can input any data, commands and so on. The execution restarts again when using the **CONTINUE** command.

```
!
!Execution of the command file continues here!
! LSS
```

CONTINUE

The **CONTINUE** command orders the engineering station to recommence the execution of the current command file after an interruption from the **USERINPUT** command. See also **USERINPUT**.

Syntax: **`CONTINUE`**

INTERACTION

The **INTERACTION** command orders the engineering station to read input data from the keyboard.

Syntax: **INTERACTION**

Description: This command makes interactive communication between the user and the engineering station possible. Input data for the command directly following **INTERACTION** is read from the keyboard. After the engineering station has received all of the input data for the command, execution of the command file continues. Use the **TYPE** command to inform the user about the required input.

Example:

This example displays at first some explanation about how to use the **DIMDB** command. The sole intention of the **INQUIRE** command is to wait until you have read all the text. The **INTERACTION** commands causes the following **DIMDB** command to accept input from the keyboard instead of the command file. Now **DIMDB** starts and you can dimension the data base as if you were working interactively. When you terminate the **DIMDB** command with **END**, the remaining lines of the command file will be processed.

```
DIMDB:
TYPE ""
TYPE "S      Expand marked segment"
TYPE "M      Modify marked line"
TYPE "!      Terminate modify mode"
TYPE "<ENTER>   Move the cursor"
TYPE ">      Cursor direction downwards"
TYPE "<      Cursor direction upwards"
TYPE "DIM    Dimension the  database"
TYPE "END    Terminate the command"
TYPE ""
INQUIRE "Press <ENTER> to continue...",dummy
INTERACTION
DIMDB
! The DIMDB picture is displayed and you can enter values.
! Terminate with END.
GOTO next
```

INQUIRE

The **INQUIRE** command asks a question. The answer is assigned to a local symbol.

Syntax: **INQUIRE** text,symbol[;**NOCONV**]

Description: The “text” parameter indicates the text which is to be displayed on the screen. This parameter follows the rules for text strings described in Section 6.5.6.1, Character Strings. Thus, this parameter must be enclosed by quotation marks.

The “symbol” parameter indicates the name of the local symbol to which the entered answer is to be assigned.

The characters in the answer are converted to uppercase, if the NOCONV parameter is omitted.

Example:

```
*!
*!List user volume’s contents
*!
*INQUIRE "Indicate user volume’s name", volume_name
*LDIR ’volume_name’:
*!
```

OLFILE

The **OLFILE** (Open Log FILE) command orders the engineering station to save all input and output in a special log file.

Syntax: **OLFILE** logfile[/**APPEND**,**READ**,**WRITE**][;**NOOUTPUT**]

Description: This command requests the engineering station to copy all input/output to the specified logfile. “Logfile” is a segment-name. The volume name can be omitted if the command is given from a command file. In this case, the name will be the same as the name of the volume in which the command file is located. If no extension is indicated, this parameter is set to “LG”.

If the segment exists, the old segment is deleted, unless /**APPEND** has been indicated. **APPEND** means that the indicated segment will not be overwritten, but rather the new data will be saved, starting at the end of the existing segment.

/**READ** means that the logfile should be opened for read (see **RLFILE**). /**APPEND** and /**READ** are mutually exclusive (/**WRITE** is default). ;**NOOUTPUT** means that only the interactive user input will be recorded in the file. Therefore the recorded logfile can be used as a command file afterwards. (NOTE: Do not forget to complete the command file with the **EXIT** command at the end!).

CLLFILE

The **CLLFILE** (CLose Log FILE) command orders the engineering station to stop logging.

Syntax: **CLLFILE**

Description: This command is used together with **OLFILE** to control the logging of input and output. **OLFILE** starts the logging, while **CLLFILE** stops it.

VERIFY

Using the **VERIFY** command, the lines in a command file can be displayed on the screen.

Syntax: **VERIFY ON|OFF**

Description: The command lines which are executed in a command file are not normally shown on the screen. **VERIFY ON** causes the lines to be echoed and makes it possible to follow the execution of the file on the screen (as well as saving it in the log file, if the log file is active). **VERIFY OFF** (normal setting) means that the command lines are not echoed on the screen.

ECHO

The **ECHO** command can be used to prevent the results of a user command from being displayed on the screen. ECHO is valid only when a log file has been opened using **OLFILE**.

Syntax: **ECHO ON|OFF**

Description: When a log file is active, the results of the user command will be saved therein, and they will also be echoed on the screen. However, this screen echoing can be suppressed by using **ECHO OFF**. **ECHO ON** is the normal setting. If system messages are presented on the same screen as the results of user commands, the **ECHO OFF** setting means that only the system messages will be displayed.

6.1.3.2 Symbols in Command Files

Symbols are used in command files as:

- File parameters
- Operands in command lines
- Variables that control execution.

Using the **EXECFILE** and **EXECSUB** commands, which start the execution of a command file, it is possible to indicate a number of parameters. The reserved symbols P1 - P9 are assigned to the given parameters, in given order. Thus, P1 assumes the value of the first parameter. All symbols are defined: those which are not given definitions are defined as empty strings.

If symbols are required in a command file, the use of local symbols is recommended, since these will be deleted after execution. This saves space in the symbol tables. Local symbols are defined using the **DFLSYMB** command (Section 6.5.3, Defining Symbols). Local symbols are typically used as parameters in commands and as variables that control execution. The expression in a symbol definition can contain symbols. A special case occurs when the same symbol is found in both symbol_name and in expression. In this case the symbol's value will be altered, since the expression is first determined and then assigned to the given symbol.
A typical application of this is a loop counter.

6.1.3.3 Commands to Control the Execution Flow

A command file is normally executed in sequence; the commands are executed one at a time, from the first to the last. You can control the execution of certain commands and specify conditions under which the execution might be discontinued. The following commands control the execution flow:

- **IF**
- **GOTO**
- **EXECFILE**
- **EXECSUB**
- **EXIT**
- **RETURN**
- **DELAY.**

IF

The **IF** command investigates the result of a comparison and executes a given command if the result is true.

Syntax:

```
IF symbol|expression comp-operator expression THEN command
```

Description: "Symbol" or "expression" specifies the name of the symbol or expression, the value of which is to be compared with the second expression. This parameter follows the rules for expressions described in Section 6.5.6, Different Expression Types to define Symbols.

The "comp-operator" defines the kind of comparison operator. Table 6-1 shows the operators which can be used. A comparison operator must be preceded and followed by a period. It cannot contain a blank character.

Table 6-1. Description of Comparison Operators

Type	Operator	Operation
arithmetic comparison operator	.EQ. .EG. .GT. .LE. .LT. .NE.	arithmetically equal to arithmetically greater than or equal to arithmetically greater than arithmetically less than or equal to arithmetically less than arithmetically not equal to
string comparison operator	.EQS. .NES.	string equal to string not equal to

The "command" parameter indicates the command which is to be executed if the result of the comparison is true.

The **IF** command makes conditional execution within a command file possible. **IF** is specially useful in conjunction with the **GOTO** command (described in the next section). Note that the engineering station distinguish between upper- and lower-case letters in comparing strings.

GOTO

The **GOTO** command continues execution at a specified line in the command file.

Syntax:

```
GOTO label
```

Description: The "label" parameter indicates the line in the command file at which execution continues. The following rules apply:

- The name of the label can consist of up to 10 characters.
- The label written in the line to continue must end with a colon (:).
- The engineering station differentiates between upper- and lower-case letters.

- Otherwise, all of the rules which apply to symbol names (Section 6.5.3, Defining Symbols) also apply to line names.

GOTO can be used in conjunction with the **IF** command to create program loops.

Example:

```
!
IF P1 .EQS. "" THEN GOTO END
LDIR
!
END:
```

In the example, execution continues at line END if the input parameter *P1* is not given.

Example:

```
!
DFLSYMB counter=0
LOOP:
```

Commands to be executed

```
DFLSYMB counter='counter'+1
IF counter .EQ. 4 THEN GOTO CONTINUE
GOTO LOOP
!
CONTINUE:
```

In the example, the “Commands to be executed” will be executed four times, then the execution continues at line **CONTINUE**.

The **EXECFILE** and **EXECSUB** commands (described in the next section) can be used for execution control by calling another command file.

EXIT

The **EXIT** command passes control directly to the command level.

Syntax: **EXIT**

Description: The command is used to pass control directly to the command level. **All** local symbols are deleted and any nested subroutine calls are skipped. See Section 6.2.2, EXECSUB - EXEcute SUBroutine.

RETURN

The **RETURN** command passes control to the calling command file. It continues at the line below the subroutine call. Local symbols defined in current file are deleted.

Syntax: **RETURN**

DELAY

The **DELAY** command generates a time delay.

Syntax:

```
DELAY delay-time
```

Description: This command makes a time delay in steps of 0.1 seconds.

6.1.4 Example

The following example shows a command file fragment using predefined symbols for command menus. First a menu is generated having a title and 8 choices for different actions. The title of the menu contains symbols and the values of them are displayed in the title. The **INQUIRE** command reads the user’s choice and saves it in symbol “answer”, which is analyzed in the following IF statements. The valid IF statement performs a jump to the according label defined behind the **GOTO** command. An illegal answer is detected in the lines below the IF statements and the menu will be reconstructed for repetition. The **EXECSUB** command (Section 6.2.2, EXECSUB - EXEcute SUBroutine) starts a subordinated command file (AADMDB.CF) on a volume defined by the symbol “cf_vol”. The parameters “targsys” and “size” are symbols and transfer input data to that command file.

```
.
.
.
MA$CLEAR
MA$MENU_0
MA$TITLE Prep of MP200 no: ’sysno’, ’systemsize’ system’s’"
TYPE ""
MA$TEXT 1. DIMDB"
MA$TEXT 2. DIMPC"
MA$TEXT 3. TRDBS"
MA$TEXT 4. TRPCS"
MA$TEXT 5. DUTDB"
MA$TEXT 6. DUTPT"
MA$TEXT 7. DUAP"
MA$TEXT 8. Make all of above actions (1-7)"
TYPE ""
MA$MENU_LINE
TYPE ""
MA$TEXT < Go to previous menu"
MA$TEXT ! QUIT"
TYPE ""
TYPE ""
MA$SAVE_POS
again_3:
INQUIRE " Make your choice: ",answer
IF answer .EQS. "1" THEN GOTO m_if_1
IF answer .EQS. "2" THEN GOTO m_if_2
IF answer .EQS. "3" THEN GOTO m_if_3
IF answer .EQS. "4" THEN GOTO m_if_4
IF answer .EQS. "5" THEN GOTO m_if_5
IF answer .EQS. "6" THEN GOTO m_if_6
IF answer .EQS. "7" THEN GOTO m_if_7
IF answer .EQS. "8" THEN GOTO m_if_8
IF answer .EQS. "<" THEN GOTO menu2mp200
IF answer .EQS. "!" THEN EXIT
TYPE ""
TYPE " Illegal answer. Try again."
TYPE " Please wait."
DELAY 15
MA$CLEAR_ANSWER
MA$RESTORE_POS
GOTO again_3
!
m_if_1:
!*************************
EXECSUB ’cf_vol’:AADMDB;"’’targsys’","’’size’"
!*************************
.
.
.
```

6.2 Executing a Command File

6.2.1 EXECFILE - EXEcute Command FILE

EXECFILE

The EXECFILE command starts the execution of a command file. The execution control is then passed to the specified file.

Syntax:

`EXECFILE` `commandfile[;parameters]`

Description: The command starts the execution of the stated command file. This command can be given from both the command level and from another command file (BATCH).

The command file parameter indicates which command file is to be executed; it is a segment identifier. If no extension is indicated, the default value CF is assumed.

If the volume name is omitted, the engineering station searches for the segment in one of the following ways:

- If the command file is started on command level, the engineering station searches for the segment in available volumes. The command file is assumed to be present in the first encountered segment with the correct name. The volume name in question is then applied.
- In the case of BATCH execution, it is assumed that the volume containing the command file which was just executed also contains the called file.

“Parameters” is used to specify a maximum of 9 parameters which are to be transferred to the command file. The parameters assign the reserved symbols P1, P2 - P9 in the order in which they are input.

Parameter 1 is assumed to be the first non-empty parameter. The parameters follow the rules for symbol definitions (Section 6.5.6, Different Expression Types to define Symbols).
The parameters must be separated by commas. Note that all of the symbols (P1-P9) are defined, since unspecified parameters are defined as empty strings ("").

How to use the Command

The following rules apply:

- Execution always starts with the first line in the command file.
- Any nested subroutine call is skipped.
- All local symbols are deleted when starting a command file.
- Execution of the command file is terminated either with the **EXIT** command, which transfers control directly to the command level, or by calling another command file (with the **EXECFILE** command).
- The only way to transfer information between two command files is via the parameters P1-P9, or by using global symbols.
- If end-of-file is encountered, an error message is issued and control passed to the command level.

Examples

Example 1:

```
*!
*EXECFILE :PROC1
*!
```

In this example, the file PROC1.CF is searched among the available volumes.

Example 2:

```
*!
*EXECFILE SRCE:PROC2.CF
*!
```

In this example, the file PROC2.CF in the volume SRCE is executed.

Example 3:

```
*!
*DFSYMB new_file="SRCE:DATA.BI"
*EXECFILE USER:PROC3.CF;"’’new_file’","ABC",10
*!
```

In the example, the command file is USER:PROC3.CF. Parameter P1 is assigned the character string SRCE:DATA.BI, P2 is assigned the string “ABC” and P3 is assigned the value 10. Parameters P4 - P9 are all assigned empty strings. The following example shows how to access the parameters in a command file:

```
!
LIST USER:'P2'.LG
!
```

In the example, the command file will list the contents of the file that has been specified as the 2'nd parameter.

6.2.2 EXECSUB - EXEcute SUBroutine

EXECSUB

The **EXECSUB** (Execute Subroutine) command makes a subroutine call of a command file.

Syntax: `EXECSUB commandfile[;parameters]`

Description: This command makes a subroutine call of the specified command file. Control is passed back to the calling file with **RETURN**. You can pass 1 up to 9 parameters to the subroutine. The command file parameter indicates which command file is to be executed; it is a segment identifier. If no extension is indicated, CF is assumed. If the volume name is omitted, the name of the default volume is taken. The parameters assign numeric or character string values to the reserved local symbols P1,P2,...,P9 in the order of entry. Parameter 1 is assumed to be the first non-empty parameter. The parameters follow the rules for symbol definitions described in Section 6.5.3, Defining Symbols. The parameters are separated with commas (,). Note that all nine symbols are defined at the start of the command file as all unspecified parameters are set to empty text strings ("").

How to use the Command

The command passes control to the specified file, and makes it possible to return to the calling file at the point of the subroutine call, by the **RETURN** command. The following rules apply for subroutine calls:

- Execution always starts with the first line in the called file.
- The maximum levels of nested calls are 8.
- Statement **RETURN** (Section 6.1.3.3, Commands to Control the Execution Flow) passes control back to the calling file, beginning with the line after the subroutine call.
- Statement **EXIT** (Section 6.1.3.3, Commands to Control the Execution Flow) passes execution control to the command level (all nested calls are cleared), and deletes all local symbols.
- Global symbols are not affected by subroutine calls.
- Local symbols in the called file are deleted at exit from it (**RETURN**).
- Local symbols in the calling file remain during execution of the subroutine.
- Symbols are searched for in the following order:
  a. Local symbol defined in current file (includes P1-P9).
  b. Local symbol defined in the calling file or caller of this, and so on.

    c. Global symbol.

- Results to the caller must be passed as global symbols.
- Actions defined with the **ON** command (Section 6.3.2, Special Procedures) are unique to each subroutine and are set to their default values at entry of it.
- If end-of-file is encountered, an error message is issued and control passed to the command level.

Examples:

```
!
EXECSUB :PROC1
!
```

In the example PROC1 is assumed to reside on the same volume as the calling file.

```
!
EXECSUB SRCE:PROC2.CF
!
```

In this example file PROC2.CF is searched for on volume SRCE.

```
!
DFSYMB new_file="SRCE:DATA.BI"
EXECSUB USER:PROC3;"’’new_file’","SECOND",1
!
```

In this example the command file is USER:PROC3.CF. Parameter P1 will be assigned the string SRCE:DATA.BI, P2 the string SECOND and parameter P3 equated to the integer value 1. Parameters P4 - P9 are all assigned the string values "".

6.3 Handling Errors

6.3.1 Error Codes

The following error codes are currently available:

Table 6-2. Currently available Error Codes

Error code	Default reaction
BREAK	continue with next line
REMCFILE	return to command level

6.3.2 Special Procedures

ON

It is possible to choose user-defined error handling, i.e. other than those executed by the engineering station. This is done with the help of the **ON** command.

Syntax: **ON** error_code **THEN GOTO** label

Description: This command defines a label where execution should continue if the specified error occurs. The specified handling applies until another handling is indicated for the same error code, or removed by the **NOON** command.

NOON

The **NOON** command cancels the user-defined error handling by **ON** for a given error code and replaces it with the engineering station’s error handling.

Syntax: **NOON** error_code

Description: This command removes the special action defined with the command **ON** and replaces it with the default action.

6.4 Predefined Symbols for Command Menus

Command menus support the user with a kind of user interface in order to input necessary data for execution or to decide which variant a command file shall run through.

The engineering station contains a number of predefined symbols to build up command menus. Those symbols begin with MA$. When creating your own menus it is not necessary to use these symbols, but they contain a number of refinements which do increase the engineering station’s performance capabilities.

6.4.1 Graphic Presentation of Menus

The menus are presented graphically on the screen with the help of predefined symbols.

MA$CLEAR	clears the screen.
MA$MENU_0	displays a solid-line frame containing 14 lines of 45 characters each. The left side of this frame is located at position 17 on the screen.
MA$TITLE	writes the title of the menu to be presented within the frame. Conclude the title with double quotes. The title appears 4 positions from the left edge of the frame.

Example:

```
MA$MENU_0
MA$TITLE Main menu"
```

This example produces the following display:

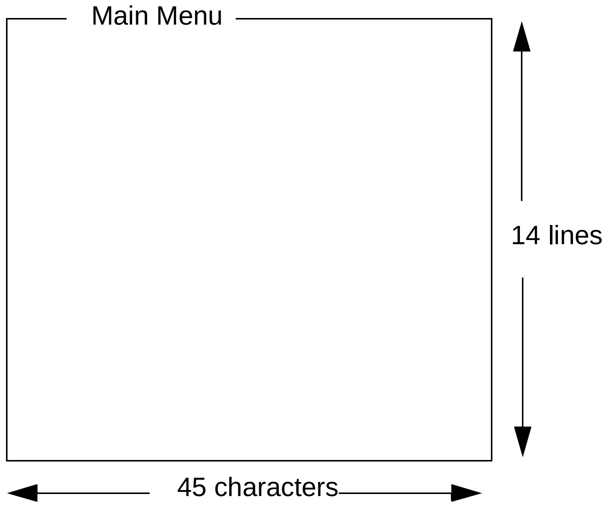


Figure 6-1. Graphic Presentation of Command File Menus

MA$TEXT displays text. The text is written out starting at position 19 on the screen, i.e., two characters from the left edge of the frame.

MA$MENU_LINE

divides the menu square into two fields. Define an empty line with: TYPE "". When creating command files, it is necessary to determine how many empty lines will be entered in order to fill the square with 14 lines.

Example:

```
           MA$MENU_0
           MA$TITLE Main menu"
(line 1)   TYPE ""
           MA$TEXT 1. Select number 1"
           MA$TEXT 2. Select number 2"
           MA$TEXT 3. Select number 3"
           TYPE ""
           TYPE ""
           TYPE ""
           MA$MENU_LINE
           TYPE ""
           MA$TEXT ! Escape menu"
           MA$TEXT < Go to previous menu"
           TYPE ""
           TYPE ""
(line 14)  TYPE ""
```

This example produces the following display:

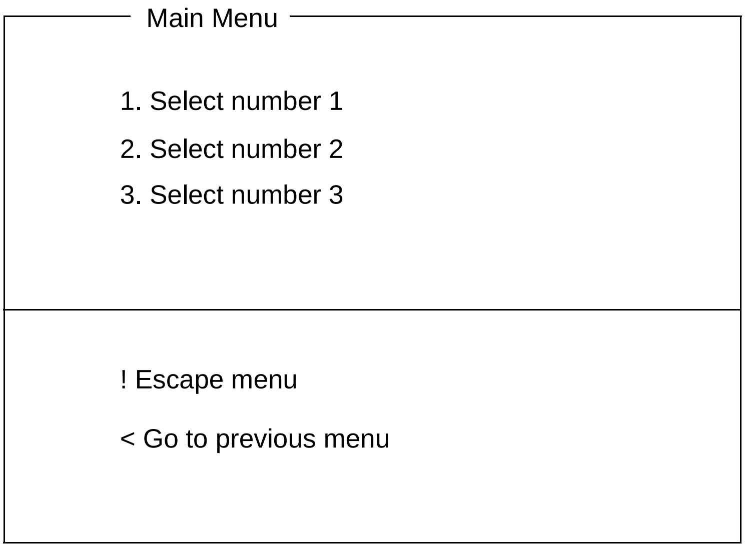


Figure 6-2. Command File Main Menu with Text

MA$UNDERLINE

draws an underline 40 characters long. The line starts at position 18 on the screen. Enter the symbol below the text you want to underline.

Example:

```
MA$TEXT Welcome to our menu"
MA$UNDERLINE
```

MA$MENU_1

up to

MA$MENU_6

are used to draw broken-line background squares to indicate where in the structure a given menu is located.

The MA$MENU_1 symbol generates a menu square which is displaced two positions to the left of the MA$MENU_0 square. Similarly, the square generated by the MA$MENU_2 symbol is shifted two positions to the left of MA$MENU_1 and so on.

MA$MENU_0 is the current menu you will make your choice from. MA$MENU_1 is the closest previous menu. MA$MENU_2 is the menu which precedes MA$MENU_1 and so on.

These symbols are specified in such an order that the current menu is displayed last, i.e. not overlapped. The specification of each menu must be followed by the MA$TITLE symbol to display the names in the background menu squares.

Example:

Assume that you have selected Menu 1 from the main menu, and then selected the current menu from Menu 1. The command file to display this structure is as follows:

```
MA$MENU_2
MA$TITLE Main menu"
MA$MENU_1
MA$TITLE Menu 1"
MA$MENU_0
MA$TITLE Menu Current"
TYPE ""
MA$TEXT 1. Select number 1"
MA$TEXT 2. Select number 2"
MA$TEXT 3. Select number 3"
TYPE ""
TYPE ""
TYPE ""
TYPE ""
TYPE ""
TYPE ""
MA$MENU_LINE
MA$TEXT ! Escape menu"
MA$TEXT < Go to previous menu"
MA$TEXT << Go to main menu"
```

This example produces the following display:

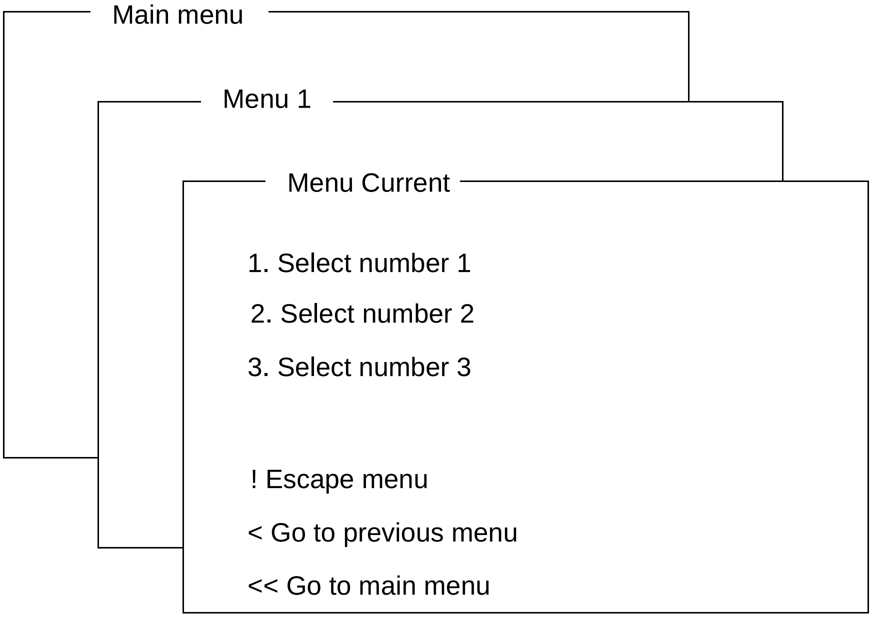


Figure 6-3. Command File Main Menu with Submenus

6.4.2 Return to specified Cursor Position

One refinement offered by the predefined symbols is the ability to return to a specified cursor position. This function is obtained using the MA$SAVE_POS and MA$RESTORE_POS symbols.

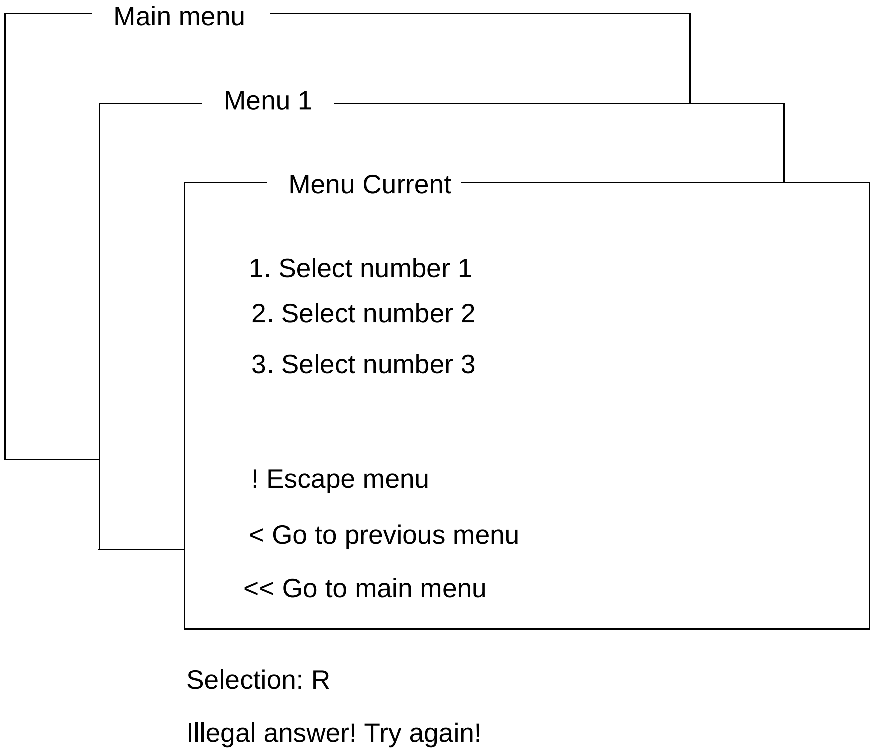
- MA$SAVE_POS saves the position of the cursor.
- MA$RESTORE_POS restores the cursor to the position saved by MA$SAVE_POS.

Example:

The following can be used in a command file with a menu as in the example of Section 6.4.1, Graphic Presentation of Menus.

```
.
MA$TEXT Go to main menu"
MA$SAVE_POS
again:
INQUIRE "Selection : ",answer
IF answer .EQS. "1" THEN GOTO xxxx
IF answer .EQS. "2" THEN GOTO yyyy
.
.
TYPE "llegal answer! Try again!"
MA$RESTORE_POS
GOTO again
.
.
```

This example leads to the following function:



```
Selection: R
Illegal answer! Try again!
```

Figure 6-4. Screen Display when giving an Illegal Answer

After “...Try again!” is displayed, the cursor will return to the position where “R” is located. You can then make a new selection.

NOTE

> In this example the display must be preceded by 18 blank characters in order to be placed under the menu.

In menus, the return to specified cursor position function can be used after a selection to return to the answer line where the selection is made. If the selection results in short displays on the screen, you see both the menu and the display from the previous selection. You can then make a new selection without having to clear the screen.

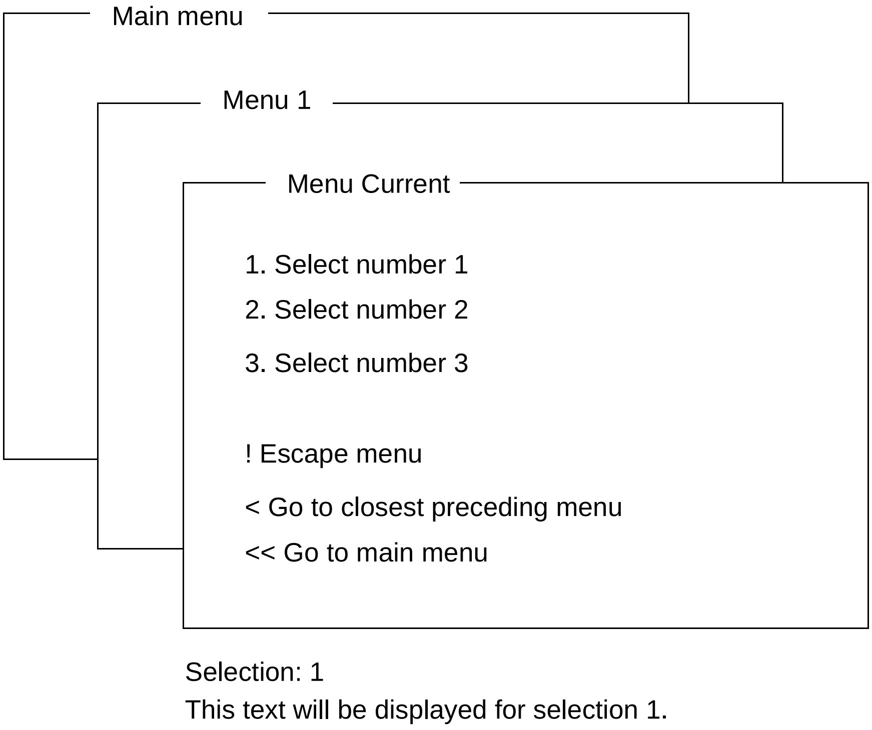
MA$CLEAR_ANSWER

> can be used to ensure that the new selection does not result in the previous display being partially written over by a new, shorter display. This symbol returns to and clears the answer from the specified cursor position, and it also clears all screen lines from the cursor position onward.

Example:

```
.
MA$TEXT Go to main menu"
MA$SAVE_POS
again:
INQUIRE "Selection : ",answer
IF answer .EQS. "1" THEN GOTO xxxx
IF answer .EQS. "2" THEN GOTO yyyy
.
.
TYPE "Illegal answer! Try again!"
MA$RESTORE_POS
GOTO again
xxxx:
MA$CLEAR_ANSWER
TYPE ""
TYPE "This text will be"
TYPE "displayed for selection 1"
MA$RESTORE_POS
GOTO again
yyyy:
MA$CLEAR_ANSWER
TYPE ""
TYPE "Selection no. 2 has been"
TYPE "made"
.
.
```

This example leads to the following function:



Selection: 1
This text will be displayed for selection 1.

Figure 6-5. Screen Display when giving a Valid Answer

The cursor will then return to the position to make the selection and clear number 1.
If selection 2 is made, the underlying text is cleared and “Selection no. 2 has been made” will be displayed on the screen.

6.4.3 Error Displays on Answer Lines

MA$ILLEGAL clears the illegal answer and causes “Illegal answer” to be displayed on the answer line 15 positions to the right of the cursor position.

The following can be entered in a command file with a menu as per the example in Section 6.4.1, Graphic Presentation of Menus.

```
.
MA$TEXT Go to main menu"
MA$SAVE_POS
again:
INQUIRE "               Selection : ",answer
IF answer .EQS. "1" THEN GOTO xxxx
IF answer .EQS. "2" THEN GOTO yyyy
.
.
MA$ILLEGAL
MA$RESTORE_POS
GOTO again
.
.
.
```

The invalid answer “R” would then produce the following screen display:

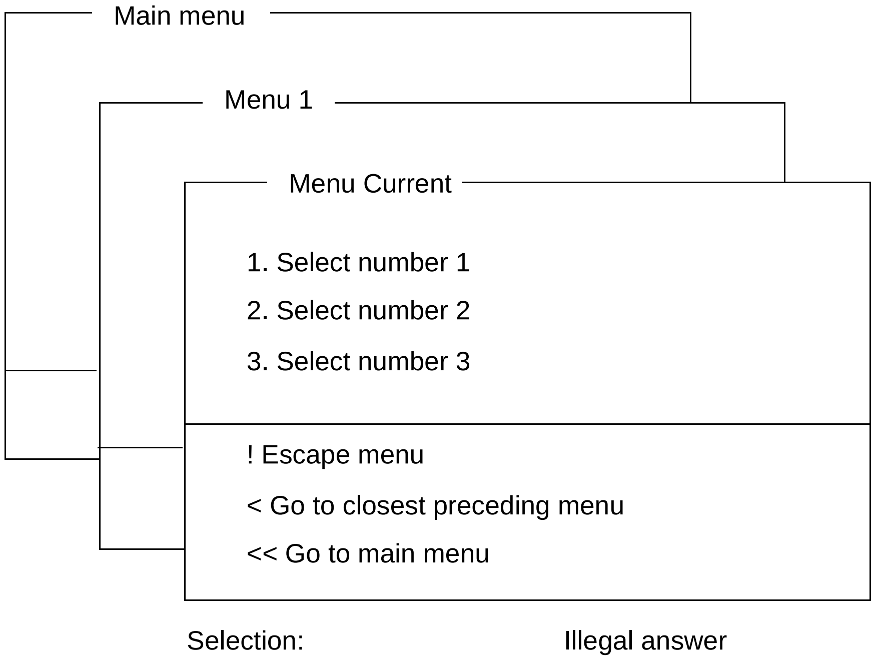


Figure 6-6. Screen Display when giving an Illegal Answer and using MA$ILLEGAL

The cursor returns to the position specified by MA$SAVE_POS, since the MA$RESTORE_POS symbol has been entered.

6.5 User-defined Symbols

6.5.1 General

Symbols can be used for the following purposes:

- As synonyms for commands/command lines.
- As variables in command files.

Symbols which are defined as synonyms for commands constitute an expansion of the command language.

A symbol which is assigned a string value and introduces a command string will be replaced by its value (see Section 6.5.6.1, Character Strings).

6.5.2 Global and Local Symbols

Symbols can be defined as global or local. A global symbol exists until the present session is terminated, i.e., until the command **TSESS** (Terminate SESSion) is given.

A local symbol is only valid within one command file and is therefore temporary. Note that local symbols may only be defined within a command file. Only global symbols are available at the command level.

Define global symbols in order to use them on command line level.

Examples:

```
*DFSYMB EX="EXECFILE SRCE:"
*DFSYMB S="SRCE:"
*DFSYMB U="USER:"
```

A set of global symbols can be easily defined by the execution of a command file, which contains the symbol definitions.

Search Order for Symbols

When a symbol is encountered, its value is searched in the following order:

1. Local symbol defined in current command file.
2. Local symbol defined in the calling file or caller of this, and so on.
3. Global symbol.

6.5.3 Defining Symbols

Global Symbol

A global symbol is defined by the command DFSYMB (DeFine global SYMBol).

Syntax: **DFSYMB** symbol = expression

Description: This command creates a global symbol. The symbol is redefined if it exists.

symbol:	name of symbol
expression:	defines the value of the symbol

“Symbol” defines the symbol’s name. The following rules apply with regard to this name:

- The name can contain letters and digits.
- The name must start with a letter.
- Upper- and lower-case letters will be treated the same.
- The only special characters permitted are the dollar sign ($) and underlining (_).
- The maximum length of symbol names is 20 characters.
- The following symbol names cannot be used (reserved for the engineering station):
  - P1, P2, ...,P9.
  - Names which begin with MA$....
  - Names which begin with F$....
  - MENU, when the engineering station is used in connection with Advant Controller 400 Series.

The different types of “expression” are described in Section 6.5.6, Different Expression Types to define Symbols.

The complete command line consists of a maximum of 132 characters. If a symbol already exists, its value is altered according to the “expression” parameter.

Local Symbol

A local symbol is defined by the command DFLSYMB (DeFine Local SYMBol).

Syntax: **DFLSYMB** symbol = expression

Description: This command creates a local symbol in current command file level’s symbol table. The symbol is redefined if it exists.

symbol:	name of symbol
expression:	defines the value of the symbol

For the definition of the symbol’s name the same rules apply as described above.

6.5.4 Displaying Symbol Values

Global Symbol

The value for a global symbol is displayed by using the **SHSYMB** (SHow global SYMBol) command.

Syntax: **SHSYMB** [symbol]

Description: This command lists the value of the specified global symbol. If the name is omitted, values of all global symbols are listed. If there is no global symbol with the given name, values of all global symbols with names starting with the given name are listed.

symbol: name of symbol

Local Symbol

The value for a local symbol is displayed by using the **SHLSYMB** (SHow Local SYMBol) command.

Syntax: **SHLSYMB** [symbol]

Description This command lists the value of the specified local symbol. The symbol is first searched in current level’s symbol table, then in upper level and so on. If the name is omitted, values of all local symbols are listed. If there is no local symbol with the given name, values of all local symbols with names starting with the given name are listed.

symbol: name of symbol

6.5.5 Deleting Symbols

Global Symbol

A global symbol is deleted by the command **DSYMB** (Delete global SYMBol).

Syntax: **DSYMB** [symbol][;**Y**]

Description: This command deletes the specified global symbol. If “symbol” is not defined, all symbols are displayed and it is asked, which of them shall be deleted. If there is no global symbol with the given name, selective deletion of all the global symbols with names starting with the given name is assumed. Selective deletion is done in query mode, unless the argument “;*Y*” is given.

symbol: name of symbol

Local Symbol

Local symbols are used only within command files. This means that all defined local symbols are automatically deleted when a command file is closed.

6.5.6 Different Expression Types to define Symbols

The value of a symbol is defined by an expression. Four types of expressions occur:

- Character strings.
- Integers and real numbers.
- Lexical functions.
- Expressions with operators.

An expression may also contain symbols.

6.5.6.1 Character Strings

A character string expression can contain characters, strings and other symbols which have been assigned character string values.

Character strings can contain any character. A character string must be surrounded by quotation marks ("). Quotation marks can be part of a character string.

A symbol can be used in a character string. Then it must be preceded by two single quotes and followed by one single quote in order to be replaced by its value. Otherwise the symbol is interpreted as a standard character string.

NOTE

> The examples in this chapter contain sometimes system prompts, sometimes not. Examples with system prompts show the interactive usage of the commands and show the according reaction of the engineering station. A command line in a command file never contains a prompt, but starts directly with the command. Exclamation points are inserted in order to improve the readability (see also Section 6.1.2, Comments).

Example 1:

```
*!
*DFSYMB seg="BACKUP.BI"
*DFSYMB short="SRCE:’’seg’ "
*!
*SHSYMB short
short="SRCE:BACKUP.BI"
```

Example 2:

```
*!
*DFSYMB g="GEPCD"
*DFSYMB p="PC2"
*!
*G ’P’
```

The resulting command line will be **GEPCD PC2**.

6.5.6.2 Integers and Real Numbers

Integer and real number expressions can contain integer values/real numbers or other symbols which have been assigned integer values/real numbers. An integer value/real number can be given in decimal or hexadecimal form. In the latter case, the number must be preceded by H’.

Example:

```
*!
*DFSYMB six=6
*SHSYMB six
six=D’6
*!
*DFSYMB fifty=H’32
*SHSYMB fifty
fifty=H’32
*!
*DFSYMB level=1.2
*SHSYMB level
level=1.2000000E+00
```

6.5.6.3 Lexical Functions

Lexical functions is a set of functions that return information about symbols. They all have names starting with F$.

The following functions are available:

F$TYPE(symbol)

The function returns a character string, depending of the type of the given symbol’s value. An empty string is returned if the symbol does not exist, otherwise one of the following is returned:

STRING	The symbol is equated to a character string in which the characters does not form a valid integer or real number.
INTEGER	The symbol is equated to an integer or a string in which the characters form a valid integer.
REAL	The symbol is equated to a real value or a string in which the characters form a valid real value.

F$LENGTH(symbol)

The function returns the length of the given symbol’s value. An error message is issued if the symbol is undefined.

Example:

```
*DFSYMB seg="BACKUP.BI"
*DFSYMB length=F$LENGTH(seg)
*SHSYMB length
LENGTH=9
```

F$LOCATE(string,symbol)

The function returns the position of the first occurrence of the given string in the given symbol's value. 0 (zero) is returned if the string does not occur within this. The string should be enclosed with double quotes. An error message is issued if the symbol is undefined.

Example:

```
*DFSYMB seg="BACKUP.BI"
*DFSYMB pos=F$LOCATE("UP",seg)
*SHSYMB pos
pos=5
```

F$EXTRACT(start_position,number_of_chars, symbol)

The function returns a sub-string of the given symbol's value. The sub-string starts at "start_position" and runs for "number_of_chars". Only valid part of the symbol's value is returned if wanted sub-part runs outside this. An error message is issued if the symbol is undefined, or any of the other parameters are less or equal to zero.

Example:

```
*DFSYMB sub =F$EXTRACT(5,2,seg)
*SHSYMB sub
sub="up"
```

6.5.6.4 How to use Lexical Functions

The main purpose for lexical functions is to check type and value of symbols.

Examples

Example 1:

Check that a given value is in the range 1-10.

```
L00:
INQUIRE "Give end-value (1-10) ",end_value
DFLSYMB symb_type=F$TYPE(end_value)
IF symb_type .EQS. "STRING" THEN GOTO L00
IF end_value .LE. 0 THEN GOTO L00
IF end_value .GT. 10 THEN GOTO L00
```

NOTE

> You can make this kind of construction more general by enclosing it as a subroutine and send upper and lower limits and possibly the question as parameters. The commands **.EQS.**, **.LE.**... are described in Section 6.1.3.3, Commands to Control the Execution Flow.

Example 2:

Check syntax of a segment name:

Volume name:	4 characters and starting with a letter.

Segment name:	Maximal 8 characters and starting with a letter.
Extension:	2 characters and starting with a letter.

```
L00:
INQUIRE "Segment (VOLUME:SEGMENT.EXTENSION) ", segment
DFLSYMB pos_colon=F$LOCATE(":",segment)
IF pos_colon .EQ. 5 THEN GOTO L10
TYPE "Volume name must be 4 characters!"
GOTO L00

L10:
DFLSYMB first_char=F$EXTRACT(1,1,segment)
DFLSYMB type=F$TYPE(first_char)
IF type .EQS. "STRING" THEN GOTO L20
TYPE "First character in volume name must be letter!"
GOTO L00

L20:
DFLSYMB length=F$LOCATE(".",segment)-6
IF length .LE. 8 THEN GOTO L30
TYPE "Length of segment name must not exceed 8 chars!"
GOTO L00

L30:
DFLSYMB first_char=F$EXTRACT(6,1,segment)
DFLSYMB type=F$TYPE(first_char)
IF type .EQS. "STRING" THEN GOTO L40
TYPE "First character in segment name must be a letter!"
GOTO L00

L40:
DFLSYMB length=F$LENGTH(segment)- F$LOCATE(".",segment)
IF length .EQ. 2 THEN GOTO L50
TYPE "Extension must be 2 characters!"
GOTO L00

L50:
DFLSYMB first_char= F$EXTRACT(F$LOCATE(".",segment)+1,1,segment)
DFLSYMB type=F$TYPE(first_char)
IF type .EQS. "STRING" THEN GOTO L60
TYPE "First character in extension must be a letter!"
GOTO L00
L60: . . .
```

NOTE

The above is not a complete syntactical check of a segment name, it just shows some examples of using the F$TYPE, F$LOCATE and F$EXTRACT functions.

6.5.6.5 Expressions with Operators

You can use operators in expressions. Table 6-3 provides an overview of allowed operators.

Table 6-3. Overview of Operators in Expressions

Type	Operator	Operation
arithmetic operators	+ - * /	arithmetic sum arithmetic difference arithmetic product arithmetic division (whole number quotient)
String operators	+ -	string linkage string reduction

Arithmetic Operators

Arithmetic operators can be used for integer calculations or calculations of real numbers. The operands can be:

- Integers.
- Symbols which have been assigned integer values.
- Real numbers.
- Symbols which have been assigned real numbers.

Example:

```
*!
*DFSYMB val_1=15
*DFSYMB val_2=10+’val_1’
*DFSYMB val_2=’val_2’*5
*!
*SHSYMB val_2
val_2=D’125
*!
*DFSYMB val_3=1.2
*DFSYMB new_val=’val_1’+’val_2’+ ’val_3’
*SHSYMB new_val
new_val=1.412000E+02
```

String Operators

String operators are used to link together or reduce strings. The operands are string values or symbols to which string values have been assigned. The plus operator (+) links two character strings together so that they form a single string. The minus operator (-) separates one character string from another.

If the string following a minus sign occurs more than once in the string which precedes the minus sign, only the first occurrence is affected.

Example:

```
*!
*DFSYMB volume="VOLB"
*DFSYMB segment="SEGB.BI"
*DFSYMB my_seg="’’volume’"+":"+"’’segment’"
*!
*SHSYMB my_seg
my_seg="VOLB:SEGB.BI"
*!
```

INDEX

A
Activities 2-1
ADDCP 2-3
Alternative choices 1-2
Arguments 1-2
Arithmetic Operators 6-30

B
Blocking on Overload 2-3
Bold strings 1-2
Bold strings in screen dialogs 1-2
Brackets 1-2

C
CLLFILE 6-5
Comma 1-2
Command Files 6-1
Command Sets 2-1
Command syntax 1-2
Commands to Control the Execution Flow 6-6
Comparison Operators 6-7
Connection via Network 2-3 to 2-5
CONTINUE 6-3
Control the Execution Flow 6-6
Conventions 1-1

D
Default Volume and Segment Names 5-2
DELAY 6-9
DFLSYMB 6-24
DFSYMB 6-24
Disconnected level (?) 5-3
DSYMB 6-25

E
ECHO 6-6
Enter key 1-2
Error Codes 6-13
EXECFILE 6-10
EXECSUB 6-12
EXIT 6-8
Expert Commands 2-3
Expert Utilities 2-3
Expression Types to define Symbols 6-26

F
F$EXTRACT 6-28
F$LENGTH 6-27
F$LOCATE 6-28
F$TYPE 6-27
Function Keys
    F1, F2, F3, F5, F6 1-5

G
GETAB
    Graphical mode 3-43
    Text Mode 3-41
Global Symbol 6-23
    Defining 6-23
    Deleting 6-25
    Displaying 6-25
GOTO 6-7
Graphical Mode 3-43

I
IF 6-7
INQUIRE 6-4
INTERACTION 6-4
Italic strings 1-2

K
Keyboard 1-5
Keyboard Interaction 1-1

L
Levels in On-line Builder 5-3
Lexical Functions 6-27
Local Symbol 6-23
    Defining 6-24
    Deleting 6-25
    Displaying 6-25

M
MA$ 6-14
Menu 2-1

N
Network Environment 5-3
NOON 6-14
Normal strings in screen dialogs 1-2

O
OLFILE 6-5
ON 6-13
Operators
  Comparison 6-7
  In Expressions 6-30
Options. See Arguments 1-2

P
Parameters. See Arguments 1-2
PC level (#) 5-3
Post Mortem Analysis 2-3
Predefined Symbols for Command Menus 6-14
Prompts 5-3

Q
Quitting Commands 1-6

R
RETURN 6-8
Right angle bracket 1-2

S
S1, S2 5-2
Screen dialogs 1-2
Search Order for Symbols 6-23
Segments 5-1
Selecting a Target 5-3
Selecting Command Sets 1-6
Semicolon 1-2
Session 2-1
SHLSYMB 6-25
Showing a Target 5-4
SHSYMB 6-25
Slash 1-2
SRCE 5-1
String Operators 6-31
Substitutions 1-2
Switches. See Arguments 1-2
Switching Running PC Programs 2-3
Symbols 6-6
Symbols for Command Menus 6-14
Symbols, user-defined 6-23
Syntax 1-2
System level (*) 5-3

T
Terminology 1-4
Text Mode 3-41
TYPE 6-3

U
USER 5-1
User defined Selection 2-3, 2-5
USERINPUT 6-3

V
VERIFY 6-5
Vertical bar 1-2
Volumes 5-1

W
Wildcards
  %, * 5-2
Working methods 5-1

3BSE 001 974R601

January 1999